UNIVERSITY OF ILLINOIS AT CHICAGO

UIC ENGINEERING
Mechanical and
Industrial Engineering

A Study on Cooperative Movement; Designing a robotic system of units which can closely mimic a collaborative method of locomotion observed in nature by insects, namely in Sawfly Larvae.

by

Kyle Kendra
Mech. E

Yuki Nojima
Mech. E

Joe Downie
Mech. E

Eric D. Najera
Mech. E

Senior Design
Fall 2021 - Spring 2022

Project Sponsor :
Prof. Pranav Bhounsule Ph.D

Project Advisor :
Prof. Matthew Alonso Ph.D

Abstract Statement

Progress requires technological innovation, which itself requires inspiration. This project is inspired by the sawfly larvae. When they travel in groups the overall movement speed is faster than that of the slowest moving larvae. The goal of this project was to mimic the way that the sawfly larvae crawl over each other like a conveyor belt and amplify the speed. To do this, three robots climb over one another and prove that the independent speed is slower than the unit speed. The end goal being to prove that there is an amplification of speed. Research and 3D modeling were the main strategies taken to pursue this project. The robots work with one another using sensors and computer programming. Multiple iterations of a tracked robot were designed. The improvements included better tolerances, build quality, and track tensioning. The final system includes three robots with tunable speeds and sensors to communicate. By measuring the time it took for one robot to travel a set distance vs the group of robots to travel the same distance; it has been confirmed that there is a speed amplification by leapfrogging robots.

Table of Contents

List of Figures, designs, and tables

Nomenclature

Symbol	Representation	Unit
W	Revolutions per minute	RPM
N	number of teeth	-
x_{1i}	Initial position of climbing robot after one second	m
x_{2i}	Initial position of carrying robot after one second	m
v_{1i}	Initial velocity of climbing robot after one second	m/s
v_{2i}	Initial velocity of carrying robot after one second	m/s
t	time	second
C	circumference	mm
D	diameter	mm

A. Problem Statement

Original Problem Description

Although sawfly larvae move relatively slow, when in a group they move fast by using each other as a conveyor belt. The goal of this project is to create a robotic system of at least two robots that copy the conveyor belt locomotion. For example, if both robots can move at a speed of v [m/s] then by using each other as a conveyor belt they can potentially move at $2v$ [m/s]. Such speed magnification through locomotion coordination has potential applications in civilian, humanitarian, and military applications such as search and rescue, transportation systems, etc.

Problem Definition:

Purpose

The UIC Robotics and Motions Laboratory has commissioned Senior Design Team #3 to observe the natural locomotion of the sawfly larva insect. Then design and construct a system of robots which mimic the speed magnification of the insects as they move together. The project is oriented to serve broader areas or research in engineering. Namely there is potential for such a project to serve applications in civilian, humanitarian, military, and transportation systems.

Objectives

The objective of this project is to produce a prototype that can demonstrate the overall increase in locomotion speed for the whole system using two or more robots. In addition, tests will be formulated and data will be extrapolated to prove or otherwise disprove the project's success in implementing the sawfly’s innate ability to coordinate a faster mode or modality.

Specifications

- Create at least two robots such that one can move over the other and vice versa. That is, use each other as a conveyer belt.
- Demonstrate that the conveyor belt idea leads to a speed amplification; the speed of the combined robots exceeds the speed of the individual robots.
- The robots can have wheels, treads, legs, or a combination.
- The size and shape can be big or small; it need not be the size of a sawfly larva.
- There is no constraint on type of actuations (e.g., electric motors, pneumatics, or others are allowed).
- The robots can be pre-programmed to move; no need for sophisticated control using sensors. Joystick control is optional.
- The systems may be tethered or untethered. A tethered system has its batteries and/or motors off-board.

B. Sponsor Background

This report consolidates the technical research findings of senior design Team #3 with the goal of developing a clear-cut database of background knowledge and sources utilized in the understanding and development of the project "*P03 ME: Sawfly Larva Insect.*" The project is sponsored by the *UIC Robotics and Motions Laboratory*. The project commissioned is that of a system of at least two robots which exhibit the apparent conveyer belt locomotion method of modality, coordinated by groups of Sawfly larva. Ultimately, it is believed that the insects are capable of coordinating a relative speed that is faster than that attainable by a single sawfly larvae's means of locomotion.

Dr. Pranav Bhounsule is currently an Assistant Professor at the University of Illinois at Chicago in the department of mechanical and industrial engineering. Dr. Bhounsule's research interests in the field includes: legged locomotion, robotics, and optimal control. He also has a github website where all his previous projects, publications are posted and has many courses that are free to anyone interested with topics ranging from dynamics, design, and robotics. Dr. Bhounsule is also the head of the Robotics and Motion Laboratory, the company sponsoring the project, and will assist the group in answering questions, concerns, and decisions that the team will make. [1]

The Robotics Lab is a research facility specializing in theoretical and applied research in robotics. Its mission is to provide a multidisciplinary environment for students and researchers interested in human-robot interaction, autonomy, and safety-critical cyber-physical systems. Current research activities range from physical human-robot interaction to safety monitoring of autonomous cars. Established in 1999, the Robotics Lab hosts students pursuing graduate degrees (M.S. and Ph.D.) in Electrical and Computer Engineering or Computer Science, as well as several undergrads. [2]

The Robotics and Motions Laboratory at UIC is interested in developing robotic prototypes and practical control algorithms that would push the state-of-the-art of current systems. We use prototyping tools such as 3D printers and simulations and tools in system identification, optimization, feedback control, and learning. Please see videos for a sampling of the systems we work on**.**[2]

This is a research project that aims to copy mechanisms in nature with the goal of publishing the research in conferences / journals. If the concept is demonstrated in prototype, then the future applications may help improve means of transporting humans and/or goods.

C. Literature Survey

Research Strategy

The strategy for research is to learn about what causes the larvae to loop over one another. Is it something that can be explained or that has been studied? Research on swarm robotics is top priority since the robots will all work with one another. To learn more about how the robots should function with one another, the nature of the sawfly larvae needs to be observed. What is the purpose of the larvae traveling in groups? Is it for an increase of movement speed, or is that merely a byproduct of them traveling together. Is this something that can be explained or that has been studied previously? Research on swarm robotics is very high on the priority list. For the robots to communicate with one another in an efficient manner then spatial organization, navigation, and decision making all need to be researched.

Biomimicry Aspect

Sawfly larvae are very similar looking to a caterpillar, but in fact they are very different. Caterpillars and Sawfly larvae have three types of legs. From front to back they have thoracic legs, abdominal prolegs, and anal prolegs. The main distinction between the caterpillar and the sawfly larvae is how many abdominal proleg pairs are observed. A caterpillar (figure 1) has 5 pairs or fewer while the sawfly larva has six pairs or more. [10]. Sawfly larvae move together as a defense mechanism against predators. They are very much a swarm mind. When alone they have a very low survival rate, but when together they stand a much better chance. Therefore, they do everything in unison. [9] The result of them traveling together has been an increase in their speed. From observation, the sawfly larvae loops over one another and they move faster as a unit than moving individually.

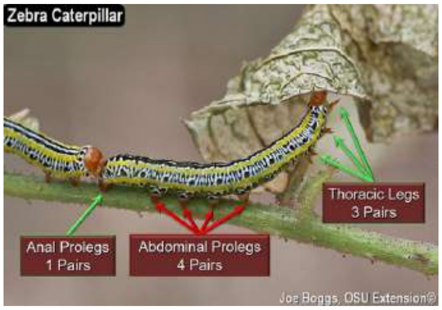


Figure 1. Features of a caterpillar larvae

Figure 2. Features of a sawfly

Figure 3. Sawfly traveling as a swarm on top on one another [4]

Swarm Robotics

Swarm robotics is when multiple robots collectively solve a problem. Swarms can get incredibly complex but part of swarm algorithms are referred to as basic swarm behaviors. This can include spatial organization, navigation, decision making, and other functions. Swarms can also be homogeneous or heterogeneous. meaning all the same robot type or a variety of robots. The reason it is a swarm is because each robot has processing capabilities, and can interact with their fellow swarm and environment. In figure 3 a tree of swarm behaviors is shown. the behaviors that we believe will be useful for the project are detailed. [13]

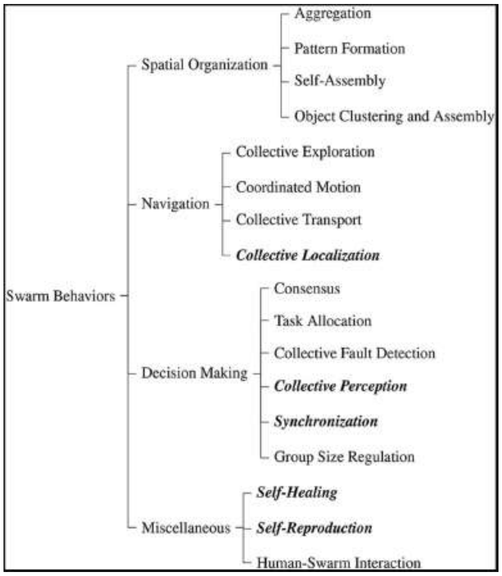


Figure 4. Swarm Robotics Behavior Trees

Spatial Organization

I. Aggregation

"Moves the individual robots to congregate spatially in a specific region of the environment. This allows individuals of the swarm to get spatially close to each other for further interaction."

This is very important because for the swarm to work in our design the individual must climb on top of one another without the ability to collectively get close to each other then there is no way for them to climb and boost each other's speed. [13]

II. Pattern formation

"Arranges the swarm of robots in a specific shape. A special case is chain formation where robots form a line, typically to establish multi-hop communication between two points."

This will be more important if we go with a swarm that has localization capabilities. But if we want to have an understanding of the formation they are currently in and if we want to optimize the efficiency of movement we would most likely want a specific formation. [13]

Navigation

I. Collective exploration

“Navigates the swarm of robots cooperatively through the environment in order to explore it. It can be used to get a situational overview, search for objects, monitor the environment, or establish a communication network.”

The project will most likely not get this developed. But if we go with a localization approach and an individual is stopped or senses an obstacle it can upload to a collective memory and form a map of the environment for the collective. [13]

II. Coordinated

“Motion moves the swarm of robots in a formation. The formation can have a well-defined shape, e.g., a line, or be arbitrary as in flocking.”

Similar to pattern formation but I believe that this is more important when it comes to movement in the z direction. When a robot is climbing it needs to know this and speed up to overcome the obstacle. Hopefully when a top speed is less important because it will already be moving with the bottom’s speed plus its own speed. [13]

III. Collective transport

“By the swarm of robots enables to collectively move objects which are too heavy or too large for individual robots.” This is also probably not going to be covered in this project. But I believe this is important for future direction and should be considered so it is possible to build on the design to reach this goal in the future. [13]

IV. Collective Localization

“Allows the robots in the swarm to find their position and orientation relative to each other via establishment of a local coordinate system throughout the swarm.” This is probably the most important swarm behavior if we want the swarm to be self-sufficient; many other behaviors rely on the localization function. Luckily, we are interested in 1 dimensional Swarm movement so there is no need for 3D localization. A few initial ideas for localization are with Dead banding, Beacon, Barcode on the side wall with a scanner on each individual robot. If we want this to be collective the Microcontrollers would need to have Bluetooth or WIFI capabilities. And we would most likely want to have a mother controller that is connected to a computer to process the information and keep track of all progress. If we do go with a Swarm with collective Localization much more detail will be given on this. [13]

Decision Making

This looks at how the robots communicate with one another. The robots collectively look at a problem and decide on an approach to overcome the problem. Through multiple methods the robots would be able to relay information.

I. Task allocation

"Assigns arising tasks dynamically to the individual robots of the swarm. Its goal is to maximize performance of the entire swarm system. If the robots have heterogeneous capabilities, the tasks can be distributed accordingly to further increase the system's performance." This is important if we go with an intelligent swarm because there will be the task of the Climber and booster. Knowing what task you have at any given moment will be important because it will boost performance. [13]

II. Collective fault detection

"Within the swarm of robots determines deficiencies of individual robots. It allows to determine robots that deviate from the desired behavior of the swarm, e.g., due to hardware failures. This is more important when we move past the 1 directional swarm movement but when we have 2 d capabilities we will need to know if an individual is left behind or veering off course. [13]

III. Synchronization

"Aligns frequency and phase of oscillators of the robots in the swarm. Therefore, the robots have a common understanding of time which allows them to perform actions synchronously." This is important because they will need to know the time if they wish to cooperate in movement [13]

IV. Human-swarm interaction

"Allows humans to control the robots in the swarm or receive information from them. The interaction can happen remotely, e.g., through a computer terminal or proximal in a shared environment, e.g., through visual or acoustic clues." This is interesting because if we are controlling the individual speed of the swarm many of the behaviors of the swarm are not needed, but I would argue that it is no longer a swarm if we are controlling the individual. But in the future direction of this project, it would be interesting to see if we could control the speed and direction of the collective swarm in real time. This is most likely the final step of this project if time wasn't a constraint. [13]

Potential design ideas

The image from the video below is a very similar project to what the group is trying to accomplish. The project above has demonstrations with up to three robots. With the demonstration with two robots, it is seen that the speed on the robot behind is increased to catch up to the robot in front. After, the robot behind catches up and the speed of the robot increases when it is on top of another robot. The robot ends up in front of the robot it was on top of and the cycle repeats. [Figure 5]

With the demonstration with three robots, a “dummy” robot was used with the motor disabled to show how a robot on top with no power will end up in front of the two motorized robots on the bottom to show that even a motor disabled robot is faster if it is on top. [Figure 6]Both demonstrations were done under ideal conditions with a flat surface and the walls providing support, so the robots do not get off course and fall off one another. [4]

Figure 5. Two robot design

Figure 6. Three robot design with a motor disabled robot included

HEXBUGs are small robotic bug toys that creep and crawl around using vibrations and were released in 2007. Throughout the years, the HEXBUG brand has released multiple different kinds of toys that work off not only vibrations, but also with motors. These newer robots also use a variety of travel mechanisms that include legs, wheels, and conveyor belts. Though there are no robots that can exactly demonstrate the increased locomotion with multiple robots, these robots can generate potential design ideas and can even be modified to be able to potentially demonstrate the main goal of the project. The pictures shown are something that the group is potentially interested in and are called the HEXBUG Fire Ant [Figure 7]. The Fire Ant are high-speed robotic bugs that can be controlled with a remote control and with some light modifications it can potentially help the group find a solution. [11]

Figure 7. Potential design ideas of a swarm shown by the HEXBUG Fire Ant and HEXBUG Larva

D. Design Specifications

Though having a robot that is efficient and/or reliable is desirable. The main focus of this project is to be able to have a working prototype, at minimum, to be able to demonstrate the core idea of increased locomotion speed with multiple robots.

I. Project Requirements

To satisfy our sponsor, the project had predetermined design requirements, needs, goals, and constraints. The criteria given are listed below.

- Cost
  The budget provided by our sponsor is up to $1200. The more cost effective the better.
- Prototype
  The group needs to create at least 2 working robots such that one can use
  another robot to amplify the robot's speed
- Speed
  The robot on top must be able to show an increase in speed compared to a robot traveling alone.
- Travel Mechanism
  The robots can use wheels, treads, legs, or a combination to be able to move.
- Size

The size and shape can be big or small; it does not need to be the size of a sawfly larva.

- Electrical Parts
  There is no constraint on type of actuations (e.g., electric motors, pneumatics, or others are allowed).
- Code
  The robots can be pre-programmed to move; no need for sophisticated control using sensors. Joystick control is optional.
- Batteries
  The systems may be tethered or untethered. A tethered system has its batteries and/or motors off-board.

II. Travel Mechanism

Caterpillar tracks work based on the same idea of a conveyor belt. They have a wide area that contacts the ground. With such a large contact area, the tracks will have better grip than a normal tire. Tracks tend to have a higher mobility compared to a tire. We intend to use a track because of the large surface area that it gives us. Robots with the track on it will have an easier time looping over one another if they have a larger surface area to work with.

E. Codes and Standards

The following is a list and brief description of codes which are currently being considered for the development of the project. Some information in these is still under revision at this point in the project. Any implementation will be described relating to the final prototype. Note that the following standards are codes which past students have used in their projects and thus became of interest to revise for our similar project :

IEEE 1872-2015: This standard defines a core ontology that allows for the representation of, reasoning about, and communication of knowledge in the robotics and automation (R&A) domain. This ontology includes generic concepts as well as their definitions, attributes, constraints, and relationships. These terms can be specialized to capture the detailed semantics for concepts in robotics sub-domains. This standard contains the Core Ontology for Robotics and Automation (CORA) with the representation of fundamental concepts from which the more detailed concepts belonging to other Ontologies for Robotics and Automation Working Group (ORA WG) ontologies are constructed. This standard also defines the ontology engineering methodology used to construct the ORA ontologies.[12]

IEEE 1625-2008:This standard establishes criteria for design analysis for qualification, quality, and reliability of rechargeable battery systems for multi-cell mobile computing devices. It also provides methods for quantifying the operational performance of these batteries and their associated management and control systems including considerations for end-user notification.

This standard covers rechargeable battery systems for mobile computing. The battery technologies covered are limited to Li-ion and Li-ion polymer, but future versions of this standard may include technologies that are not in general use at present.

II. Technical Content

F. Assumptions

Ideal conditions.

For this project we are assuming that the surface we are working with will have no debris of any kind for the robots to have to overcome. The surface will be smooth for the robots to be able to drive over. There will be walls enclosing the robots so they cannot go far from one another and will not stray outside of the demonstration zone. There will be no airflow pushing the robots around or potentially flip over the robots.

Materials

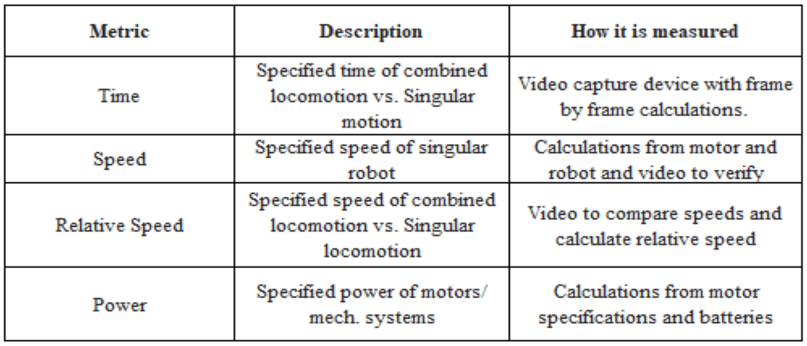
For the project the group is set on a restricted budget for the whole project. The project should not use any material that would cause the project to go over budget.

Time

The time limit for this project is towards the end of the spring semester. Time organization should be monitored using the GANTT chart so that the team does not fall behind when it comes towards any deadline. With a time constraint, the project should be presentable with a working prototype before building on more ideas.

G. Metrics

For this project, many metrics need to be measured to be able to observe if the target final goal or design criterias are met. The results of the metric will determine whether or not the final design is an appropriate design.



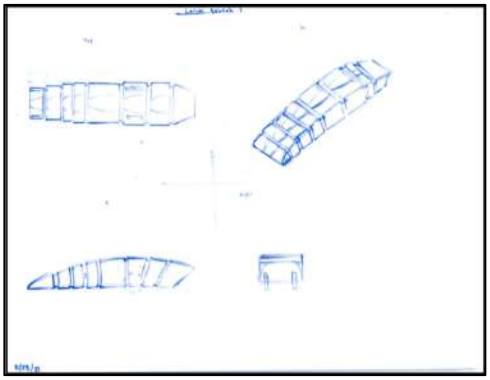
Metric	Description	How it is measured
Time	Specified time of combined locomotion vs. Singular motion	Video capture device with frame by frame calculations.
Speed	Specified speed of singular robot	Calculations from motor and robot and video to verify
Relative Speed	Specified speed of combined locomotion vs. Singular locomotion	Video to compare speeds and calculate relative speed
Power	Specified power of motors/ mech. systems	Calculations from motor specifications and batteries

Table 1. Metrics Table

H. Proposed Solutions

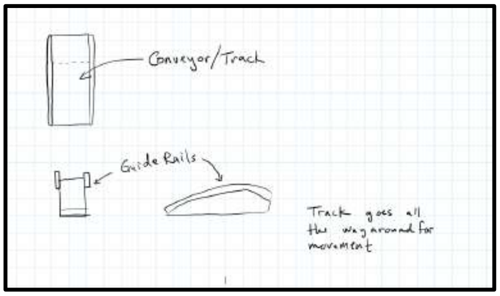
The designs below are the designs the team came up with that have the potential to reach the final goal of the project. The designs will be ranked based on the decision matrix table.

Design 1: “Wheel Bug”



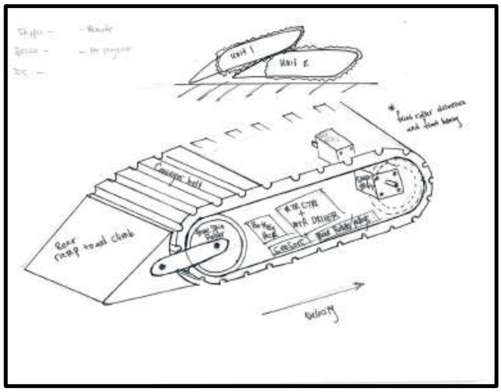
Design 1 is a wheel driven robot. The overall shape is oriented to ease the climb of one robot over the other. It has an incline on one end and curves to allow for a descend in front of the leading robot. The design borrows from an existing robot toy *“Hexbug Larva Toy”* See Figure 7. but is based on a redesign of the general body. Namely the shape is flatter and wider than the existing toys. This design is most closely resembling a larva since it adapts a small amount of motion that makes it resemble the insect physically. This design avoids the complexity that may come with integrating pulleys or gears in the design but this design may not provide the most speed increase if it moves on two wheels only.

Design 2: “Track Bug”



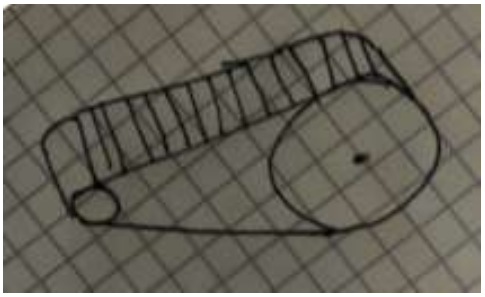
Similar to the first design, but with the addition of bumpers in the up climb over each robot integrated to the top.

Design 3: “Single track bot without an incline”



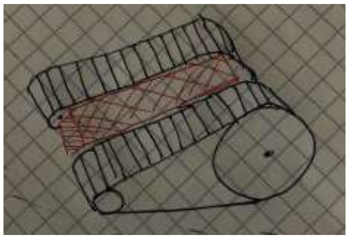
This design implements a single track with no incline. Since there is no incline, an added ramp would need to be attached to help the climbing robot go over the carrying robot. The robot being flat adds to the stability for the climbing robot, but there are still concerns with the width. This design is very simple, which is nice for a large swarm of robots to naturally boost each other's speed.

Design 4:(Single with incline)



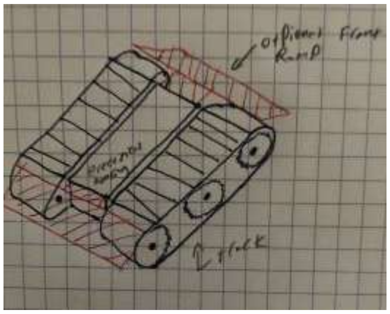
This design uses a single incline. This incline would be caused by the front gear being larger than the back gear. Since the track creates a ramp, the climbing robot would get an immediate assist in going over the carrying robot. There is some concern that it may be unstable for the climbing robot to go over the carrying robot. Another consideration would be where to house the components needed to move the robot.

Design 5:(Double with incline)



This design implements two tracks on an incline. This incline would be caused by the front gear being larger than the back gear. Since the track alone creates a ramp the climbing robot would get an immediate assist in climbing the carrying robot. Since there are two tracks this will add width to the robot further stabilizing the climbing robot. This will also allow for a middle section that can house other components.

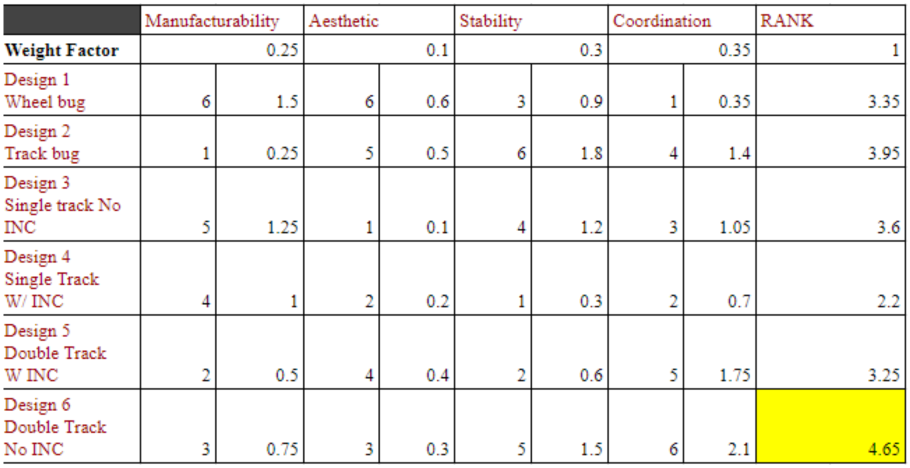
Design 6:(Double without incline)



This design implements 2 tracks that have capabilities to go different speeds or direction or have the same speed and movement. It has a ramp on the tail end to assist robots with climbing on top of one another. and an optional front track that will help with descent for the climbing robot. but the front track may cause issues with climbing so we will need to run tests to see how it performs. one other key aspect of this design is that the gears within the tracks will all remain constant. and there will also be a section in between tracks that has the capabilities to house components.

I. Selected Design with Decision Matrix

As a team the weight factor, ranking of each design, and the criterias results were all decided. After the calculations it was determined that Design 6, Double Track No Incline, has the highest score out of the designs. Manufacturability represents how easy the design would be to manufacture. Solidworks designs, material used, and previously existing designs that could be bought and repurposed were all used to weigh in on this portion. Aesthetics were based on how much the design resembles a sawfly larva and what the group thinks of the design. For stability, the main factors considered were width of the robot, guide rails, as well as incline and decline of the robot. This is to ensure that the climbing robot doesn't fall off one of the edges and to keep the robots from flipping over. Coordination is about how well the robots can climb over one another. This includes being able to speed up quick enough to leave a short time gap before the robots reach one another again, the incline of the robot to see how easy it would be to climb one another, the amount of surface area that the tread/wheel would cover, as well as the grip it provides. Once the robots climb over one another the roles change. The speeds of the robots will switch after the climbing robot completely dismounts the carrying robot. This will be included in the coordination part of the decision matrix.



	Manufacturability		Aesthetic		Stability		Coordination		RANK
Weight Factor	0.25		0.1		0.3		0.35		1
Design 1 Wheel bug	6	1.5	6	0.6	3	0.9	1	0.35	3.35
Design 2 Track bug	1	0.25	5	0.5	6	1.8	4	1.4	3.95
Design 3 Single track No INC	5	1.25	1	0.1	4	1.2	3	1.05	3.6
Design 4 Single Track W/ INC	4	1	2	0.2	1	0.3	2	0.7	2.2
Design 5 Double Track W INC	2	0.5	4	0.4	2	0.6	5	1.75	3.25
Design 6 Double Track No INC	3	0.75	3	0.3	5	1.5	6	2.1	4.65

Table 2. Decision Matrix

III. Methodology

J. Preliminary Calculations

The principle of relativity states that “all systems of reference are equivalent with respect to the formulation of the fundamental laws of physics.” [5] When riding on a train you are moving at the same speed as the train. When you start walking in the same direction as the train at 1 mph, then you are moving faster than the train. Say the train is going 20 mph and you are going 1 mph. Now you are going 21 mph. The same concept applies to this project. Consider two robots stacked on top of each other. If the bottom robot is moving 1 mph, then the climbing one is also moving at 1 mph. Then if the climbing robot begins to move 1 mph while still on top of the other moving robot, it will be moving at 2 mph. This is the rough idea for how this project will work. When one robot loops over the bottom one, the speed of the climbing robot will be amplified. Then when the robot falls off and the robot that was previously on the bottom loops over, its speed will be amplified. As a whole, the system of robots should be moving faster than 1 mph of a single robot.

K. Calculations

- **I. Equations**

The following equations were used for the calculations and conversions from inches to mm.

$$1\ in = 25.4mm \tag{1}$$

$$W_2 = N_1 * W_1 / N_2 \tag{2}$$

$$C = D\pi \tag{3}$$

$$Speed\ of\ Robot = C * \mathrm{W} \tag{4}$$

$$x_f = x_i + v_i t + \frac{a_i t^2}{2} \tag{5}$$

$$\frac{(v_{1i} + v_{2i}) + v_{2i}}{t} \tag{6}$$

The following data represents rough ideas of what speeds, gear ratios, motor specs, and dimensions would be needed to get the robot to a working point that is functional and meets the goal of this project.

II. Velocity and position switch

To get a better understanding for how much faster the robots will travel when working together, a python script was created. The python script assumes ideal conditions for each case. The first case being two robots crawling over one another, and the second being three robots crawling over each other. For the first case, the robot coming from behind will climb up the leading robot and travel over. Once the climbing robot dismounts, the robots will switch roles and repeat the process. This is assuming that there is no stalling from the extra weight of the robot, that the switch is instantaneous without a gap between the two after the one robot has dismounted the other. For the second case, the robots will be in a line. For this example the robots will be in positions 1, 2, and 3 with 1 being the furthest behind. Robot 1 will climb robot 2 and as soon as it passes robot 2, robot 2 will speed up and climb up robot 3. In this way, the robots' motion is almost circular while still traveling further. Again this assumes all ideal conditions.

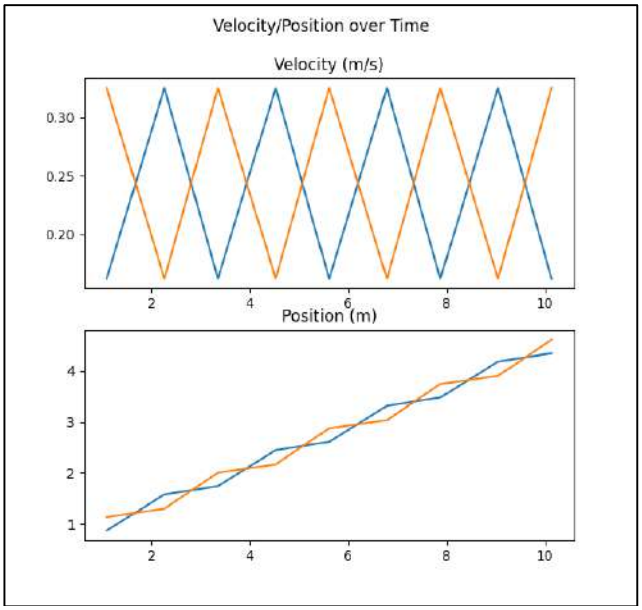


Figure 8: Graph showing the velocity changes and the position changes for two robots

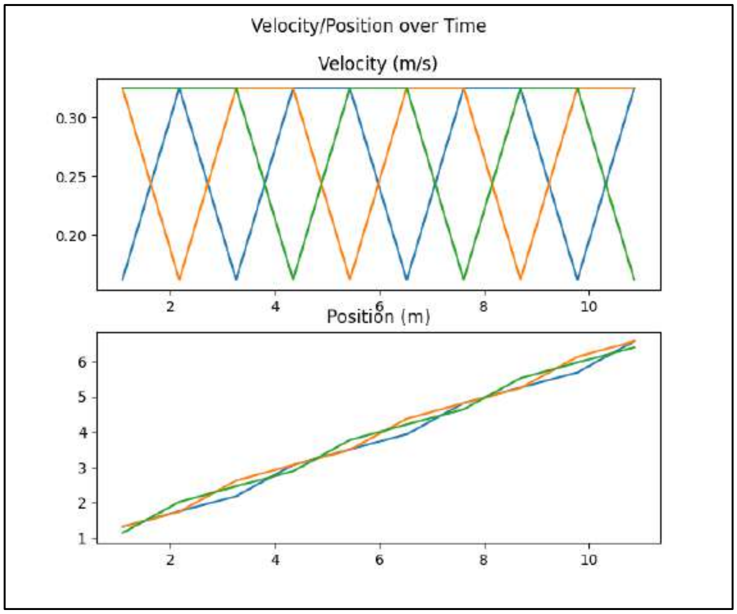


Figure 9: Graph showing the velocity changes and the position changes for three robots.

Distance (m)	Time (s)		Percent Time Difference (%)		Speed (m/s)	Percent Difference (%)
3.13	9.6	1 Robot	0		0.32	0
				AVG	0.32	0
3.48	7.9	2 Robots	20%		0.44	30%
3.74	7.9		20%		0.48	38%
				AVG	0.46	34%
3.50	5.4	3 Robots	56%		0.64	66%
3.49	5.4		56%		0.64	66%
3.77	5.4		56%		0.69	72%
				AVG	0.66	68%

Table 3: Speed differences from one robot moving at its maximum speed to two and three robots crawling over one another.

The graph shows how the robots switch positions and velocity based on their position. Each color represents a different robot. The speeds and distances are placeholder based on robot size and actual speed. Looking at Table 3, it can be seen that an ideal speed percentage increase would be about 38% for two robots, and 68% for three robots.

III. Gear Size relation to robot speed assuming 200 RPM motor

Teeth Gear	diameter (inches)	diameter (mm)	Circumference (mm)	RPM	Robot speed (mm/min)	robot speed (m/min)	robot speed (m/s)
30	3.90	99.06	311.21	200	62241.23	62.24	1.04
24	3.16	80.26	252.16	200	50431.36	50.43	0.84
18	2.43	61.72	193.91	200	38781.08	38.78	0.65
12	1.69	42.93	134.86	200	26971.20	26.97	0.45
6	0.97	24.64	77.40	200	15480.51	15.48	0.26

Table 4. Gear size vs Speed

From this data we discover the speed of an individual robot given the amount of teeth or the outside diameter of the driving gear. as the driving gears circumference increases so does the speed. all speeds are based on the gear being connected to a motor that rotates at 200 revolutions per minute.

IV. Gear speed for a given amount of teeth and assuming 200 RPM on Gear 1

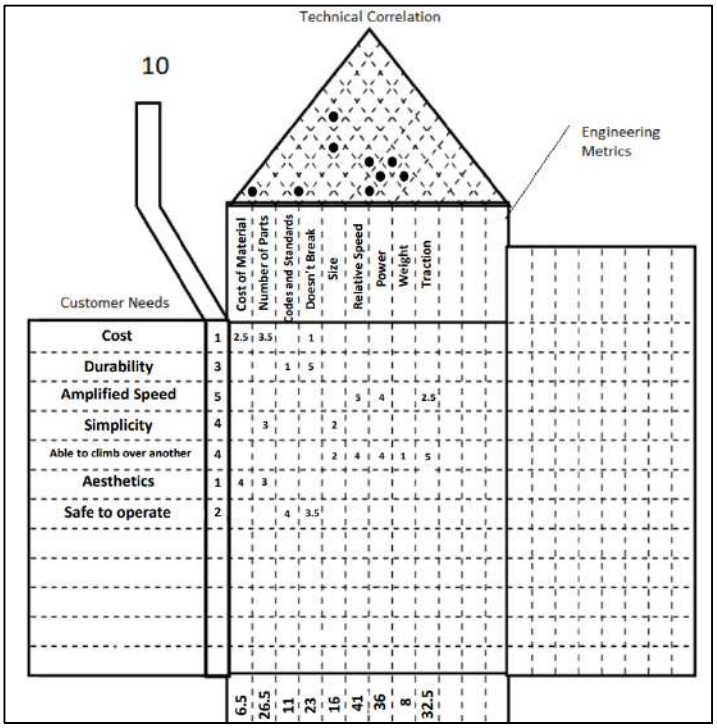
Teeth Gear 1	**Teeth gear 2**	**Gear ratio**	**W1 (RPM)**	**W2 (RPM)**
30	30	1	200	200
30	24	1.25	200	250
30	18	1.67	200	333.33
30	12	2.5	200	500
30	6	5	200	1000
24	24	1	200	200
24	18	1.33	200	266.67
24	12	2	200	400
24	6	4	200	800
18	18	1	200	200
18	12	1.5	200	300
18	6	3	200	600
12	12	1	200	200
12	6	2	200	400
6	6	1	200	200

Table 5. Gear ratio vs Speed

Given the amount of teeth of the driving gear, gear 1. then you can find the RPM of the second gear given you know the number of teeth on gear 2. and the RPM of gear 1. The equation being $N1/N2=W2/W1$ W2 would also be negative if the gears were in direct contact but since they will be interacting with a belt they will rotate in the same direction.

P. House of Quality

A QFD (Quality Function Deployment) is an important part in the decision making process. It is able to show the customer request and requirements and translate them into a system where the most important attributes are ranked at the bottom. The higher the number the more important the attribute is. Looking at the rankings it can be seen that the number of parts, relative speed, and traction are the most important attributes that need to be thought about when designing and will be thought about.



Customer Needs		Cost of Material	Number of Parts	Codes and Standards	Doesn't Break	Size	Relative Speed	Power	Weight	Traction
Cost	1	2.5	3.5		1					
Durability	3			1	5					
Amplified Speed	5						5	4		2.5
Simplicity	4		3			2				
Able to climb over another	4					2	4	4	1	5
Aesthetics	1	4	3							
Safe to operate	2			4	3.5					
		6.5	26.5	11	23	16	41	36	8	32.5

Figure 10. Quality Function Deployment

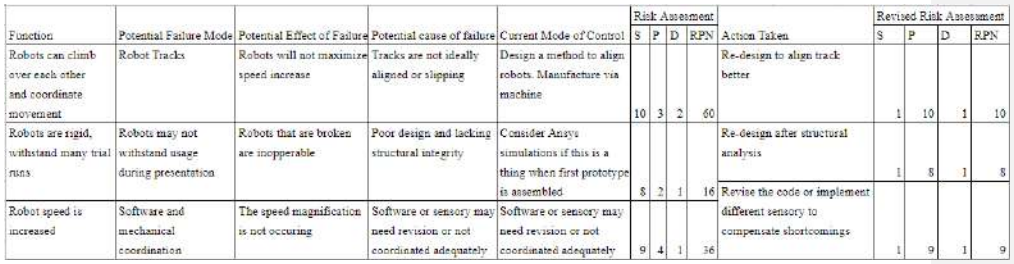
IV. Preliminary Results

Q. FMEA

The first function that we aim to maintain is for the robot's ability to climb over one another. The tracks and how well aligned these are is a big factor that determines the robot's ability to achieve the maneuver. Therefore, this is of high importance. The probability of this occurring is relatively low because the robots should be easy to assemble. As a mode of control we will square the robot and measure properly to align and be able to maximize the potential speed achieved.

The second function depends on the rigidity of the robots. The robots are not meant to last long term but should withstand through the prototyping phase and well into the final week of final design expo. If the robots do not have a good build quality, then there will be issues with making more. The probability of this occurring is also low. Our mode of control is to use adequate materials and hardware.

The third function that we aim to achieve is the speed magnification via coordination. The severity if this function is not achieved is high as well. There is a relatively higher chance that we encounter issues with this as we don't know yet how the system will behave until we prototype a real version. Also the programming or coordination is yet to be developed further.



					Risk Assesment					Revised Risk Assessment			
Function	Potential Failure Mode	Potential Effect of Failure	Potential cause of failure	Current Mode of Control	S	P	D	RPN	Action Taken	S	P	D	RPN
Robots can climb over each other and coordinate movement	Robot Tracks	Robots will not maximize speed increase	Tracks are not ideally aligned or slipping	Design a method to align robots. Manufacture via machine	10	3	2	60	Re-design to align track better	1	10	1	10
Robots are rigid, withstand many trial runs	Robots may not withstand usage during presentation	Robots that are broken are inopperable	Poor design and lacking structural integrity	Consider Ansys simulations if this is a thing when first prototype is assembled	8	2	1	16	Re-design after structural analysis	1	8	1	8
Robot speed is increased	Software and mechanical coordination	The speed magnification is not occuring	Software or sensory may need revision or not coordinated adequately	Software or sensory may need revision or not coordinated adequately	9	4	1	36	Revise the code or implement different sensory to compensate shortcomings	1	9	1	9

Figure 11. FMEA from the project

V. Preliminary Conclusions / Future Work

Problem with Design 6 from Design Matrix

While doing a motion study of if the desired design from the design matrix would work, Design 6, an issue arose while observing the motion study. It was noticed that the fixed front ramp would collide with the hinged back ramp and could cause malfunctions overall. Another thing that was considered was the possibility of when the robot is descending from the leapfrog mechanic, the front ramp could collide with the ground and can also cause failures. With all considered, it was decided to remove the front ramp unless it was absolutely necessary.

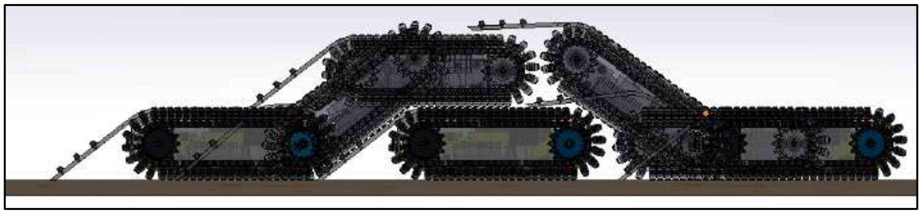
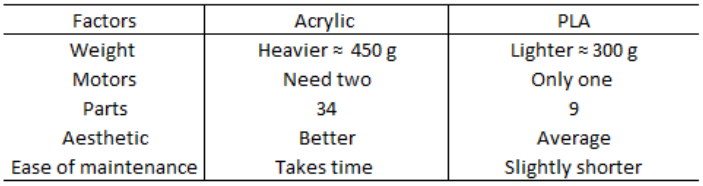


Figure 12. Motion study of the robot with just the back ramp

Prototyping

With design 6 without the front ramp becoming the design choice for this project, the design material was one of the biggest considerations that had to be made. From the group's discussions two material choices were made, the first is 3D printing with PLA and the other is a combination of 3D printed parts with the main body consisting of laser cut acrylic parts. After both prototypes were made some information was able to be compared. The table below shows some of the pros and cons of Acrylic vs PLA.



Factors	Acrylic	PLA
Weight	Heavier ≈ 450 g	Lighter ≈ 300 g
Motors	Need two	Only one
Parts	34	9
Aesthetic	Better	Average
Ease of maintenance	Takes time	Slightly shorter

Table 6: Pros and cons of Acrylic vs PLA

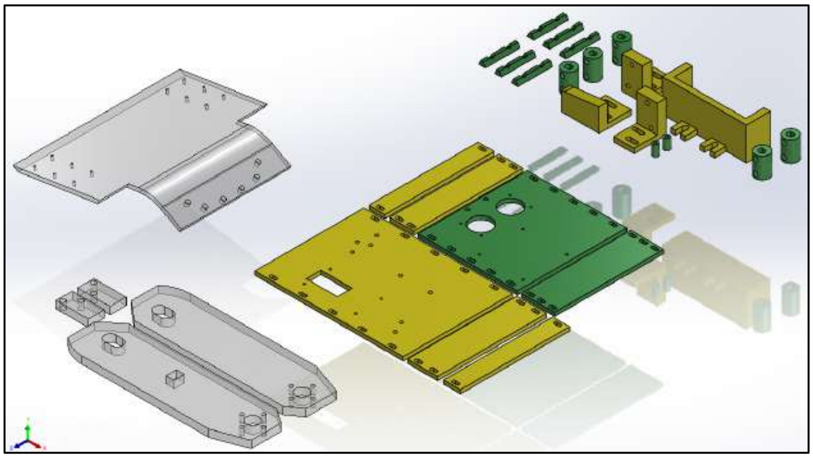


Figure 13: Acrylic Laser cut design parts

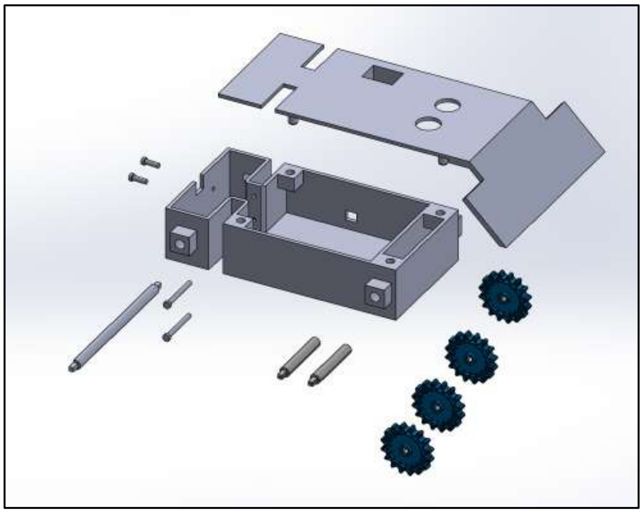


Figure 14: PLA 3D print parts

Looking at the figures above just in terms of parts alone, the 3D printed design would have less parts than the acrylic parts. Even if more components were added to the 3D print and/or parts were removed from the laser cut design, the 3D print should overall have less parts meaning less sources for errors.

The acrylic part also has more parts such as the ramp supports and parts to secure the housing better. The acrylic design also has a ramp and hinge feature where the ramp can freely move up and down. From the motion study from Figure 12, it can be seen that the front ramp would have contact with the ground and having a fixed ramp would cause problems, so the idea was decided to be implemented to the 3D printed design as well.

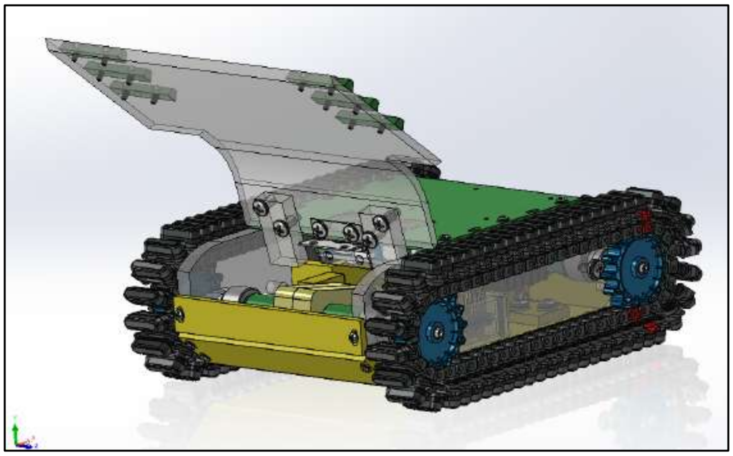


Figure 15. Acrylic design showing how the ramp and hinge would work.

Another thing that had to be considered are the potential errors and mistakes that can occur during the design process. Our group believes that before going into extensive laser cutting, everything should be 3D printed and tested before going into the acrylic design. The first priority is to accomplish the goal of the project, being able to show an amplification in speed by using a leapfrog mechanic, and if the 3D print design is successful and time remains, the acrylic should be the next step. All Prototype drawings can be found in the Appendix.

Multiple iterations of prototypes

With the makerspace at the team's disposal we quickly started 3d printing chassis to house components ordered. The initial thought was to have 3d print to work on code and arduino wiring. While we worked on the arduino another design consisting mainly of laser cuts assembled together would house electrical components. The team initially believed that the Laser cut method would be stronger and more reliable. As we continued to 3D print multiple iterations it became clear to the team that 3d prints could be reliable and can be manufactured faster and cheaper. Because of the speed that 3d print prototypes can be manufactured, adjustability is not a concern, instead the focus is to find errors in design and correct them on the next prototype till a design that works consistently is found.

Prototype A

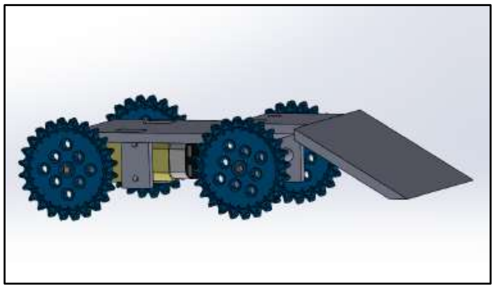


Figure 16. Prototype A Solidworks Assembly

Prototype A was designed using the 24 tooth gear in mind. With a simple arduino system at this time there was no concern for housing the electrical components with the belief that the electrical system would expand. The focus of this design was to get the motor working and simply go forward. Problems found with this design was the width of the bed. The gears are larger than the laser cut design which was going to use 16 tooth gears. This design did prove that we can move forward at a constant speed.

With this design we had no axis made of more permanent components so axes to be 3d printed were designed.

The design with 2 squares on both ends is the rear axle to be connected to 2 gears on both ends is named Gear to Gear.

The Design with square insert on one end and oval cut out on the other end is to connect the motor to the front gears.

Figure 17. Gear to Gear A

Figure 18. Motor to Gear A

The issue with this design is that it can not easily be printed. So the axis were redesigned before printing to a design that would have a flat face flush with the quare insert that connects with the gears.

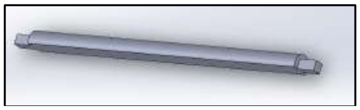


Figure 19. Gear to Gear B

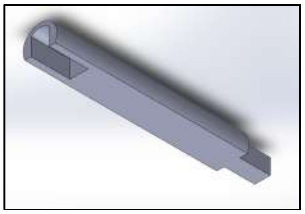


Figure 20. Motor to Gear B

This design was found in 3d print very well. The Motor to Gear axis did slip off of the motor often so the part was taped to the motor which corrected the slip off issue.

Prototype B

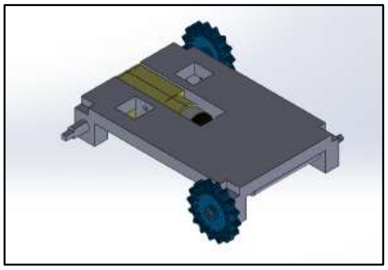


Figure 21. Prototype B Solidworks Assembly

Prototype B was Printed with the focus to have electrical components sit more neatly on the top of the bed. Transitioning to 16 tooth gears cut outs for tread clearance were added. Issues with this design was that electrical components would still fall off and wires would get tangled in the track. There was also no possibility to have another robot travel over this design

Prototype C

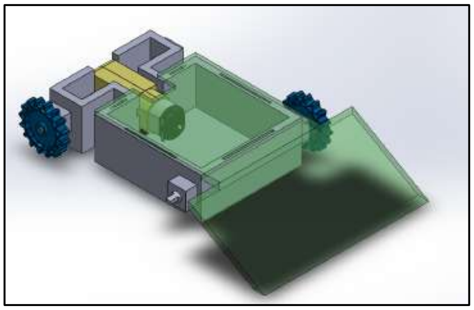


Figure 22. Prototype C Solidworks Assembly

This design we transitioned to a design with capabilities to enclose electrical components. The issues with the design was that with an expansion of the arduino system there was no room for all the arduino components. So the rear must be expanded. Another issue found was the height was not tall enough to enclose all components so the height was increased. After printing and assembling the connectors on the lid / ramp were too weak. It was also found that Taping the axle to the motor was not reliable anymore with the axel being enclosed.

Prototype D

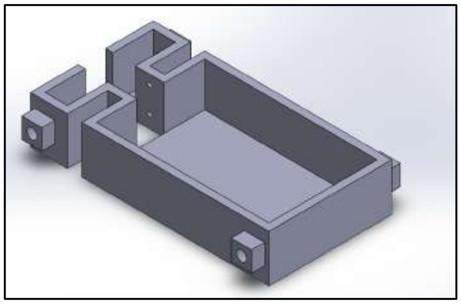


Figure 23. Prototype D Solidworks Part

Prototype D was never printed; the design was strange and was abandoned to be reworked into Prototype E. The issue of the Height was never fixed and the way to connect the lid to the chassis was never fixed.

Prototype E

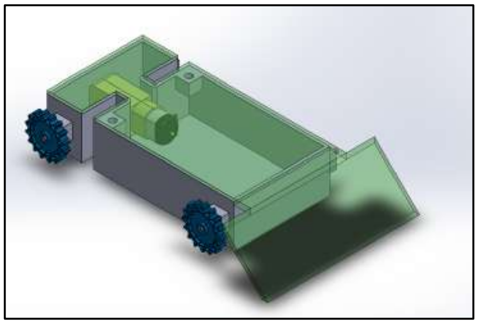


Figure 24. Prototype E Solidworks Assembly

This was one of the most functional designs. This design proved that we could 3d for the final design. All electrical components had room. No cutout in the bottom or lid was needed for the motor to fit in. We also tested if the walls would be strong at 2.5 mm thick instead of 5 mm thick. Which was successful. Issues with this design was that the Lid would move around when the robot was in motion. The track was colliding with the top of the ramp. The wires would also get tangled with the Gear to Gear Axle. With the final design closing in, the issue of track tension is now a concern with this design having a loose track.

To correct the problem of not being able to reliably tape the front axle to the motor 2 parts are made to be glued together and become one motor to gear axle. The issue found was that the smaller side was not reliably printed.

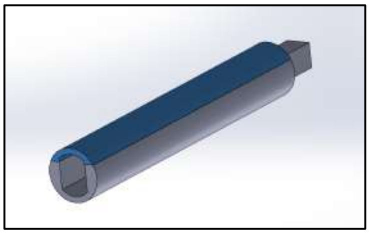


Figure 25. Motor to Gear E

Prototype F

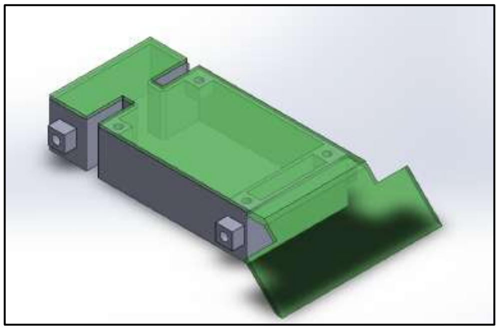


Figure 26. Prototype F Solidworks Assembly

In prototype F the cut out for the Gear to Gear axle was moved back 2 mm which was found to be too much with an over tensioned track. The ramp had an additional cut out that gave clearance to the track. A wall was added to separate the electrical housing and the Gear to Gear Axle. Another 2 pegs were added to the lid to ensure that the lid is connected well. Issues with this design was the over tensioning of the track. The need for cutouts in the Lid for the ultrasonic sensor and front for the proximity sensor. We also found it annoying to take the lid off to disconnect the battery.

To correct the issue of the axle 2 equally sized parts were printed to be glued together. We found that this is a reliable method and no issues have been found with this design.

Figure 27. Motor to Gear F

Prototype G

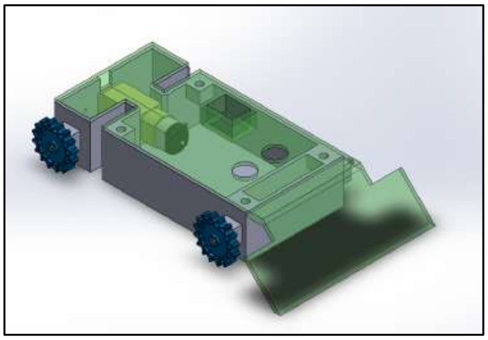


Figure 28. Prototype G Solidworks Assembly

In Prototype G Cut outs for sensors were added also with the purchase of a power switch a place for that was added into the design. The Cut out for the rear axle was moved forward to correct tensioning issues. After Testing with Prototype G we found that a hinge should be added that connects the ramp to the lid so the Ramp does not interfere with descent. This was the final adjustment and the next design was the final design.

Prototype G minor changes

With the Motor to Gear part breaking often, it was thought that adjusting the part would be a quality of life change. This would save time and cost when the part just breaks and testing cannot be done on that day and additional parts need to be printed. The figures below shows the new design that can be inserted with the wheel axis that came with the kit.

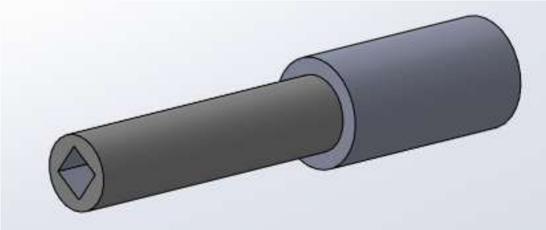


Figure 29. CAD drawing of the Motor to Gear G

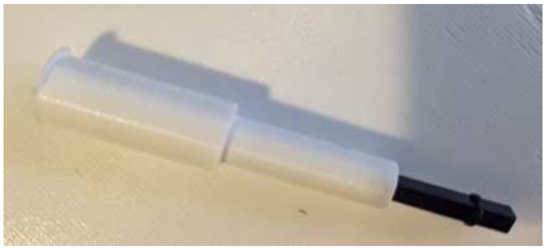


Figure 30. Part printed with the axis attachment

Commented [1]: add figure #

Track Design

To get consistent testing results we designed a track for the robots to be tested in. The track consists of a track wall with .25 inch plywood .25 clear acrylic and a connector. To cut the walls the laser cutter at the UIC makerspace was utilized with a DXF file. The connector is built with a vex robotics aluminum structure kit. using 2x2x35 Aluminum Angle pieces (276-2304) cut so that there are 5 holes across. and a 5x25 Aluminum Plate piece (276-2311) cut so that it has a length of current iteration of the robot. connect with nut and bolt with rounded end facing inward on the track. Below are a picture of the DXf (Figure 31) and a picture of the track in use (Figure 32). as you will see in figure 24 the clear acrylic makes it very easy to observe the robots.

Figure 31. Track wall DXF

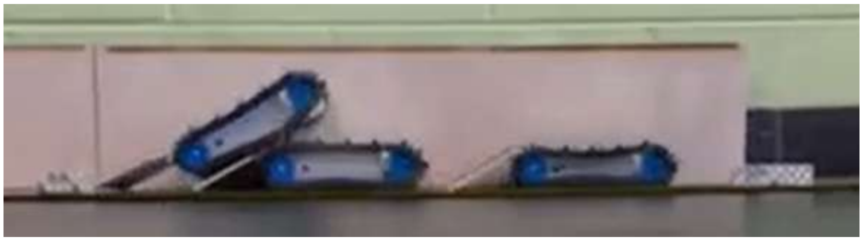


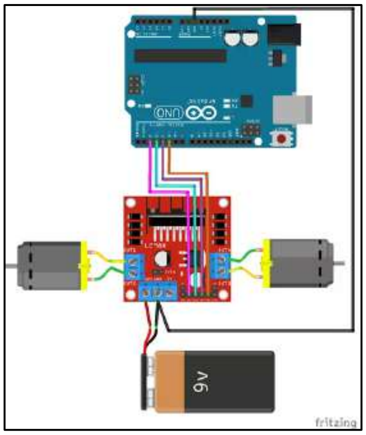
Figure 32. Track in use

VI. Results

L298N H Bridge Motor Driver vs. Transistor

Both designs accomplish a similar result, the robot is able to move and is able to adjust its speed using the ultrasonic sensor, but there are some issues with the transistor setup shown below. The transistor setup has trouble controlling the amount of voltage and current that will go into the motor. The motors that are used in the project are at full potential when the motor is at 6 V and around 160 mA. Without a voltage regulator or anything to control the voltage from the 9 V battery it will eventually fry the motor after prolonged use. If the system is powered through the arduino and everything is powered through the 5V supply pin, the motors will not reach 6 V and will not reach the current needed because each arduino pin can only output 40 mA, so the maximum torque for the motor can not be obtained. The transistor setup can work, but it will require much more housing inside the robot, increasing its size. [14]

With this in mind, the L298N bridge solves all the issues from above by reducing the amount of wires needed and being able to reach the voltage and current needed for the motor to fully function. It is able to output up to 2A of current and has an integrated 5V power regulator so the ultrasonic sensors can fully function as well. The only downside of the L298N is the height of the heat sink, measuring at approximately 27 mm. This height was not accounted for at the start of prototyping and adjustments were made after prototype E. [15]



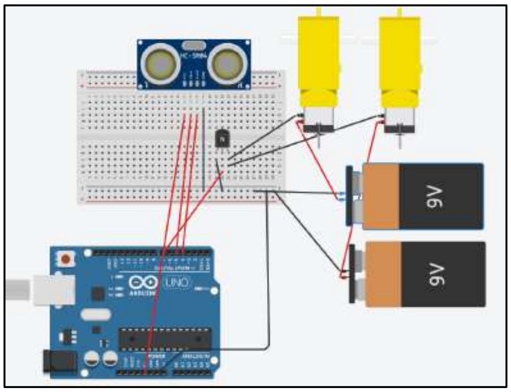


Figure 33. An example electric schematic of the L298H motor drive, on the left, and the NPN transistor on the right.

One or Two motors?

With the L298N motor drive being used, the option of one or two motors can be implemented with ease. The only downside with using two motors is that the 9V battery would not have enough voltage and current for the motors to function at full potential. To solve this, a 12V power supply will be used instead, which is around 50% larger than the 9V battery, which will also take up more housing inside the robot. The results from this testing would determine if the size of the housing needs to be increased for the extra motor and the larger battery needed for the design.

From the prototypes made by using 3D printing, tests were conducted to see whether or not the robot will need one or two motors to climb an approximately 30 degree slope that will eventually be implemented into the design. From just testing the one motor setup with the 9V battery, it can be seen below that the robot is capable of climbing the 30 degree slope and a two motor setup did not need to be tested.

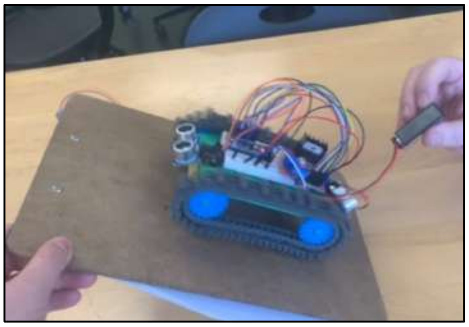


Figure 34: A demonstration showing that Prototype B with one motor was able to climb approximately a 30 degree incline

Sensors

Initially, there was only one sensor implemented on the top of the robot, shown in Figure 33 with the transistor. The purpose of the sensor is to make the robot recognize that the robot behind was able to leapfrog over itself and realize it was behind the robot that just leapfrogged. This setup would work in a two robot project where the robots are just going over one another, but in a system of more than two robots the one sensor robots will fail eventually. This is because even if the robot was able to go over one another, there could be slight errors where the landing force could cause the robot to distance itself more than the robot can travel while it is on top of one another. To minimize the distance from robot to robot, a front sensor was added so it will be able to maintain a specific distance from robot to robot. This would reduce the possibility of the robot falling between a space and messing up the order of the robot.

Figure 35. Assuming the rectangles are robots, the figure above shows that the three robot design can fail if the space between the two robots that the robot behind is trying to leapfrog over has too much distance between one another.

Final Design

Commented [2]: Don't add this for midterm?

Commented [3R2]: I am sure that we have to

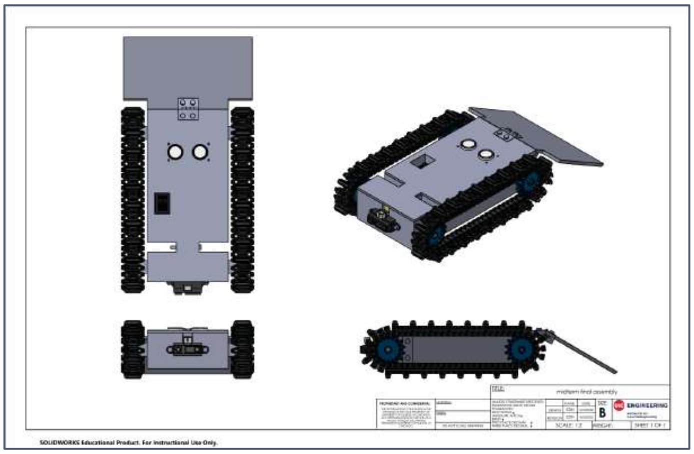


Figure 36. Prototype G PLC Solidworks assembly

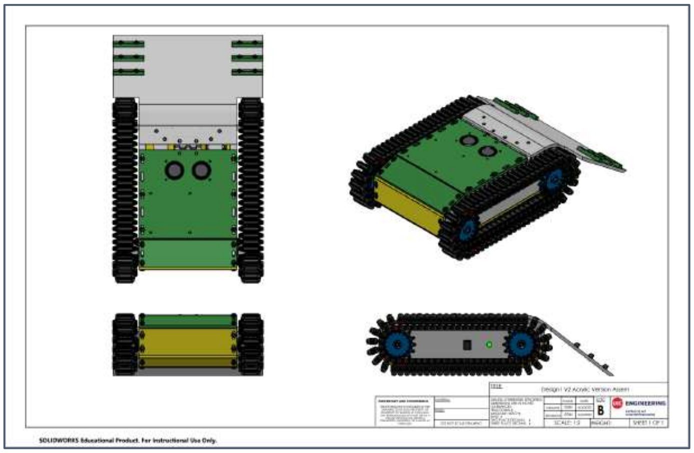


Figure 37. Acrylic with PLC assembly

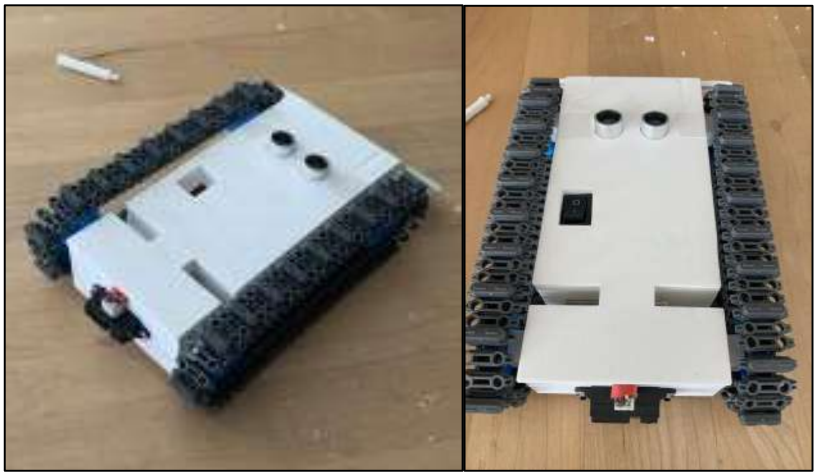


Figure 38. Robot Final Design

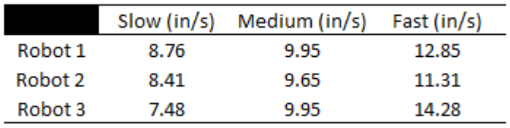
At the time of this report Prototype G has become the final design. This Design can be easily assembled taking about 5-10 minutes to put the pieces together after parts are completed. This design has all the useful components from the previous designs that include: the power switch, sensor holes, motor to gear axis, hinges for the ramp on the back. Three Prototype G robots will be assembled for the testing that will need to be done.

Individual speed differences

Individual speeds were measured using the track seen in Figure 31 and 32. When initial measurements were being taken, it was observed that the robots traveled at different speeds even with the same build. This is because each robot is unique in a sense where the slight differences in build and the electrical components can cause uncertainties to occur from robot to robot, and a variable needed to be introduced to slow down the robots if the robot was catching up at a time where it was not supposed to. This is only an issue when three or more robots are traveling together and the middle robot can potentially overtake the robot that is in front due to having a speed higher than the robot in front of it. To compensate for the different speeds, code was implemented where a certain distance needs to be maintained between the robot in front and the robot in the middle.

Commented [4]: change

For this experiment three speed settings were used, slow, medium and fast. The slow setting comes into effect when the middle robot needs to slow down in case the robot in front is moving at a slower speed, the medium speed is the average speed the robot travels for example when the front most robot is just traveling, and the fast speed was used for the robot that is farthest behind to catch up to the robots in front. Speeds for each of the robots was collected by taking a video of the robot traveling individually. Using a video editor, the exact time it took for the robot to travel the whole entire track was measured. This was conducted three times for each of the speeds, with all the robots. The table below shows the data for each of the robots for each speed.

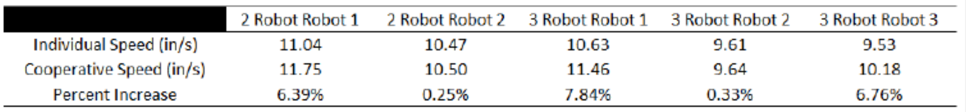


	Slow (in/s)	Medium (in/s)	Fast (in/s)
Robot 1	8.76	9.95	12.85
Robot 2	8.41	9.65	11.31
Robot 3	7.48	9.95	14.28

Table 7. Speed of each robot at the three speeds.

Cooperative Speeds

The purpose of this project was to find an amplification in speed when the robots moved as a group. The method on how the speeds were collected was identical to when the robots were traveling individually. Because the track is short, the robots can only go over one another once or twice depending on the position of the robot. The speed changes for each of the robots was also measured by using a video editing tool and observing when the speed increased or decreased and used the corresponding individual speed to find the speed as if the robots were moving individually. The individual speed and the cooperative speeds were compared in the table below and it was found that there was a speed increase in all robots overall. It was observed that Robot 2 in the two robot setup and Robot 2 from the three robot setup had a lower percent increase compared to the other robots. The reason it happens is because the robot came to a halt when a robot climbed over it causing the speed to decrease to 0 for a very short time. This is most likely caused by the motor not having enough torque to carry itself and the robot on top.

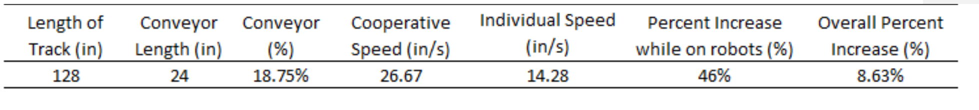


	2 Robot Robot 1	2 Robot Robot 2	3 Robot Robot 1	3 Robot Robot 2	3 Robot Robot 3
Individual Speed (in/s)	11.04	10.47	10.63	9.61	9.53
Cooperative Speed (in/s)	11.75	10.50	11.46	9.64	10.18
Percent Increase	6.39%	0.25%	7.84%	0.33%	6.76%

Table 8. Speed of robots moving in cooperative motion and the percent increase observed.

Ideal max speed increase

Max speed increase occurs when the distance between the robots that are being climbed over are close enough that the robot going over can travel over the robots without falling between the robots. Data was collected in the same manner as the individual and the cooperative robot speeds. The table shows the following, the length of the whole track, the length of the conveyor belt, the conveyor length compared to the track, the cooperative speed when the robot is on top, the individual speed if it was not traveling cooperatively, the percent increase while on the robots, and the the overall percent increase on the track. Figure 39 below shows the ideal setup when the robots are traveling. A 46% increase in speed was measured when the robot was able to successfully travel over both robots without falling.



Length of Track (in)	Conveyor Length (in)	Conveyor (%)	Cooperative Speed (in/s)	Individual Speed (in/s)	Percent Increase while on robots (%)	Overall Percent Increase (%)
128	24	18.75%	26.67	14.28	46%	8.63%

Table 9. Max speed of robot observed

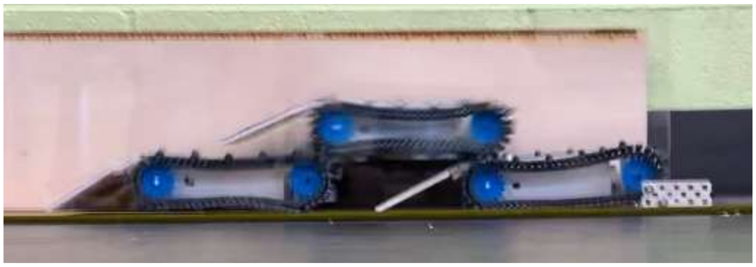


Figure 39. Ideal speed increase for the three robot movement

Future Direction

Some issues were found when tests were conducted such as the halting when a robot is being climbed over and the distance from one robot are not close enough at times where an ideal speed increase could not occur. To solve these problems, two motors with a bigger battery could be used for the speed. The only problem with that is the space inside of the robot. The battery will need to be adjusted to a 12 V battery due to having two motors and because there is an extra motor inside of the robot, more space will be needed, a new body will need to be 3D printed. Robots that can communicate with each other by wireless communication are also an option to maintain constant distance from one another. Other parts of the robot can also be adjusted are the diameter of the wheels to increase the potential speed and a different motor. All the things mentioned here and other options could improve the project overall.

Lessons learned

The most important lesson learned from this class is managing time and thinking ahead of time. Some of the tasks that were planned on the GANTT chart took much longer than anticipated and had to be adjusted because every member of the team had other things to work on.

Statement of individual contributions

Kyle - Research, programming, recording meetings, prototyping, GANTT, decision making,

Yuki Nojima - Electronic schematic, arduino code, prototyping

Joe Downie - Solidworks design of prototypes and rapid prototyping, identify problems with design and redesigning for next prototype iteration.

Eric Najera - Solidworks designing, part ordering/suggestions,

Conclusions

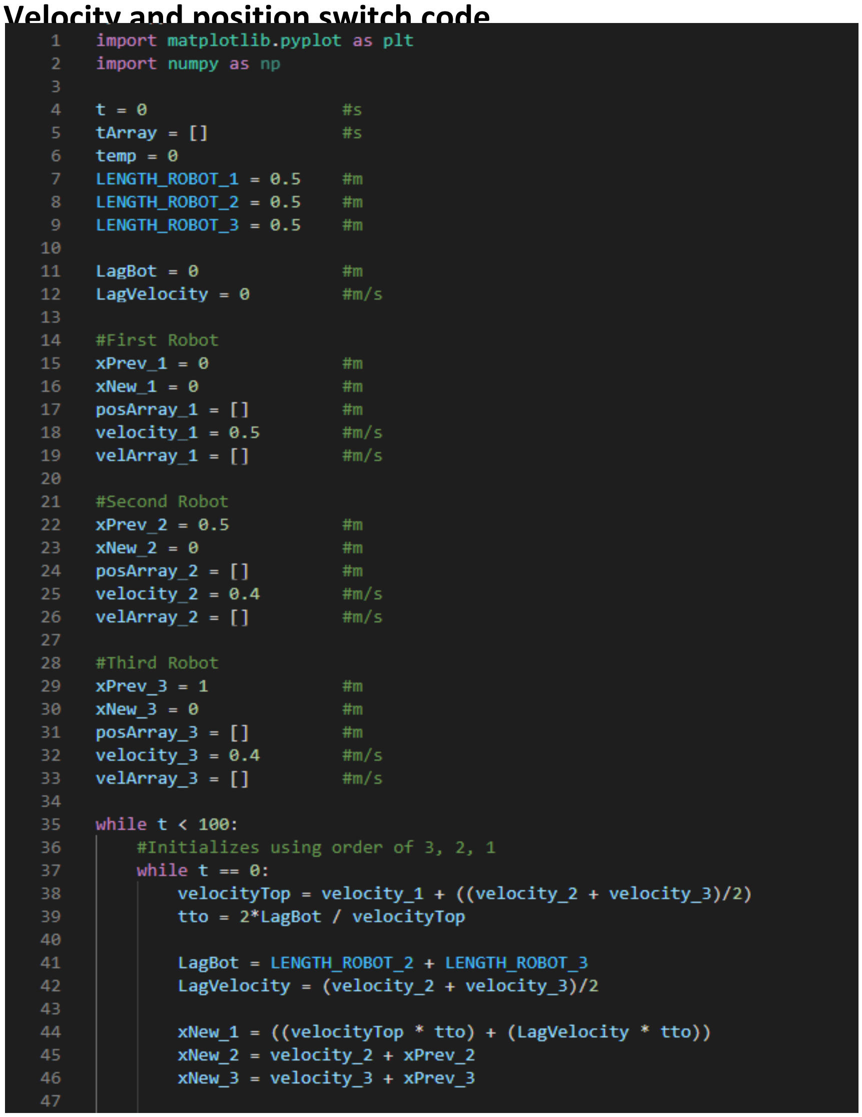
An increase in speed when robots are moving in cooperation was able to be observed. This was true for both the two robot setup and the three robot setup. With an adjustment in speed being introduced to the robots depending on the position of the robot, the group was successfully able to mimic the movement of the sawfly larvae effectively. Theoretically, the increase in speed could be much higher if more robots were made. Also, to further improve on this project, more motors can be added to the robots with a bigger battery to be able to have the torque and power to withstand the weight of the robots that are top of the robots.

Appendix

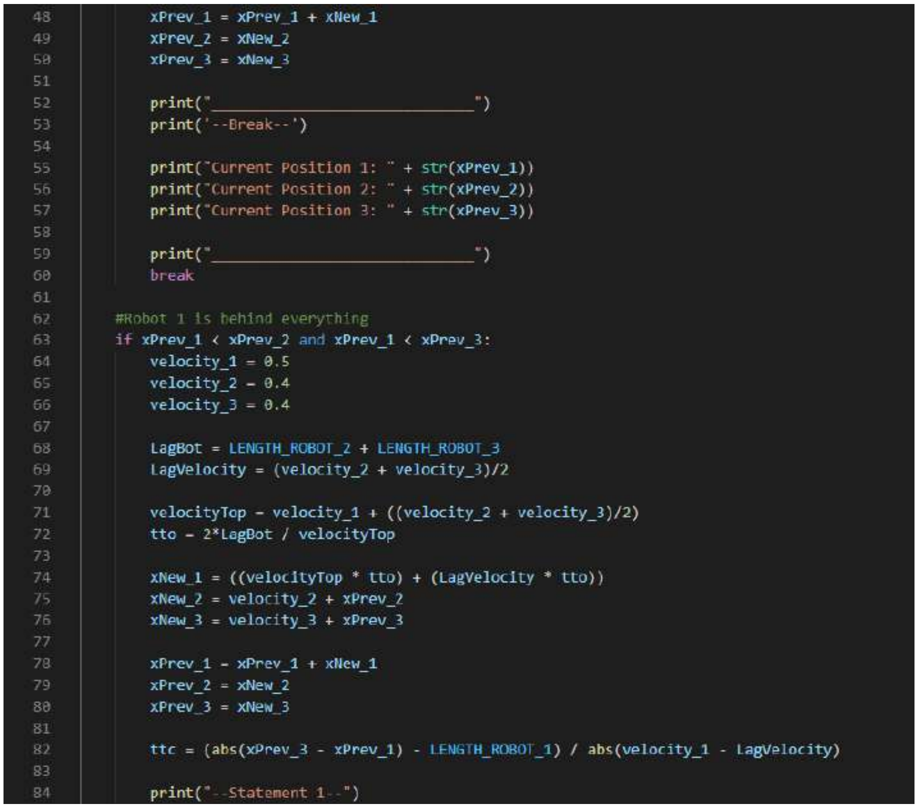
Final Design BOM

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.	Unit Price ($)	Extended Price ($)
1	Chasis	3D printed (100g)	1	$0.03/g	$3.00
2	DC Motor		1	$2.95	$2.95
3	M3x45mm		2	$0.10	$0.20
4	Gear to Gear G	3D printed (3g)	1	$0.03/g	$0.09
5	L298N Motor Driver		1	$2.50	$2.50
6	Rechargable 9V battery		1	$7.01	$7.01
7	ARDUINO NANO		1	$6.66	$6.66
8	MiniBreadboard		1	$0.45	$0.45
9	hc sr04 ultrasonic sensor		1	$1.30	$1.30
10	16 T Wheel		4	$0.25	$1.00
11	RA1113xxx1 (ON/OFF Switch)		1	$0.90	$0.90
12	Chain		120	$0.06	$7.20
14	Rubber Traction Link		18	$0.10	$1.80
15	Long Range SHARP IR Sensors - #GP2Y0A21YK0F		1	$7.00	$7.00
16	M3x5		10	$0.10	$1.00
17	0.1 m to g	3D printed (2g)	2	$0.03/g	$0.06
18	Lid Flap	3D printed (35g)	1	$0.03/g	$1.05
19	Lid	3D printed (40g)	1	$0.03/g	$1.20
20	Hinge		1	$0.50	$0.50
21	Vex Robot Connection		2	$0.10	$0.20
				Total Price	$46.07

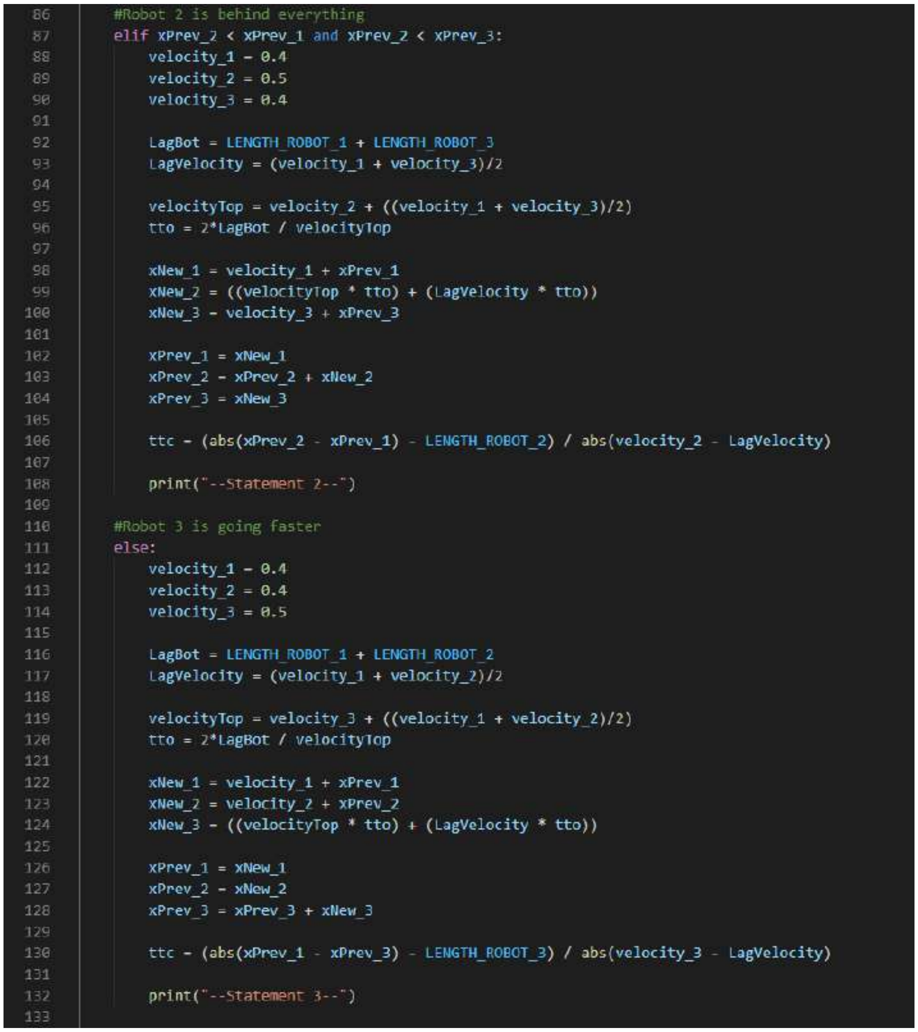
Velocity and position switch code



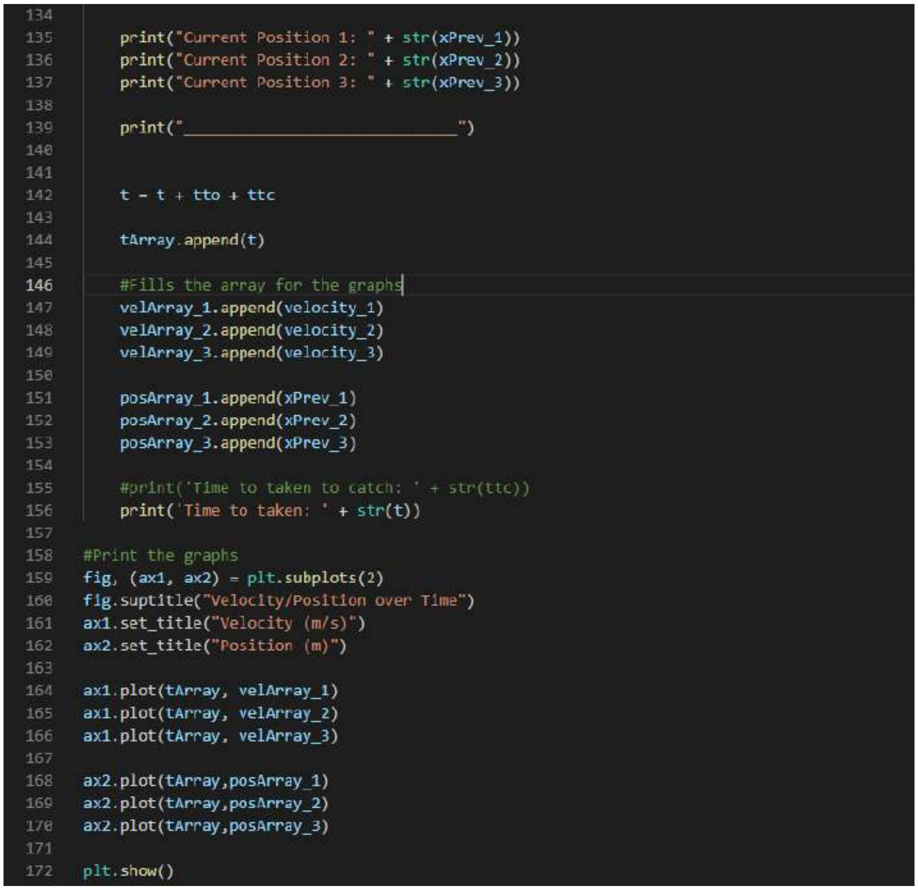
```python
import matplotlib.pyplot as plt
import numpy as np

t = 0                   #s
tArray = []             #s
temp = 0
LENGTH_ROBOT_1 = 0.5    #m
LENGTH_ROBOT_2 = 0.5    #m
LENGTH_ROBOT_3 = 0.5    #m

LagBot = 0              #m
LagVelocity = 0         #m/s

#First Robot
xPrev_1 = 0             #m
xNew_1 = 0              #m
posArray_1 = []         #m
velocity_1 = 0.5        #m/s
velArray_1 = []         #m/s

#Second Robot
xPrev_2 = 0.5           #m
xNew_2 = 0              #m
posArray_2 = []         #m
velocity_2 = 0.4        #m/s
velArray_2 = []         #m/s

#Third Robot
xPrev_3 = 1             #m
xNew_3 = 0              #m
posArray_3 = []         #m
velocity_3 = 0.4        #m/s
velArray_3 = []         #m/s

while t < 100:
    #Initializes using order of 3, 2, 1
    while t == 0:
        velocityTop = velocity_1 + ((velocity_2 + velocity_3)/2)
        tto = 2*LagBot / velocityTop

        LagBot = LENGTH_ROBOT_2 + LENGTH_ROBOT_3
        LagVelocity = (velocity_2 + velocity_3)/2

        xNew_1 = ((velocityTop * tto) + (LagVelocity * tto))
        xNew_2 = velocity_2 + xPrev_2
        xNew_3 = velocity_3 + xPrev_3

```



```
        xPrev_1 = xPrev_1 + xNew_1
        xPrev_2 = xNew_2
        xPrev_3 = xNew_3

        print("______________________________")
        print('--Break--')

        print("Current Position 1: " + str(xPrev_1))
        print("Current Position 2: " + str(xPrev_2))
        print("Current Position 3: " + str(xPrev_3))

        print("______________________________")
        break

    #Robot 1 is behind everything
    if xPrev_1 < xPrev_2 and xPrev_1 < xPrev_3:
        velocity_1 = 0.5
        velocity_2 = 0.4
        velocity_3 = 0.4

        LagBot = LENGTH_ROBOT_2 + LENGTH_ROBOT_3
        LagVelocity = (velocity_2 + velocity_3)/2

        velocityTop = velocity_1 + ((velocity_2 + velocity_3)/2)
        tto = 2*LagBot / velocityTop

        xNew_1 = ((velocityTop * tto) + (LagVelocity * tto))
        xNew_2 = velocity_2 + xPrev_2
        xNew_3 = velocity_3 + xPrev_3

        xPrev_1 = xPrev_1 + xNew_1
        xPrev_2 = xNew_2
        xPrev_3 = xNew_3

        ttc = (abs(xPrev_3 - xPrev_1) - LENGTH_ROBOT_1) / abs(velocity_1 - LagVelocity)

        print("--Statement 1--")
```

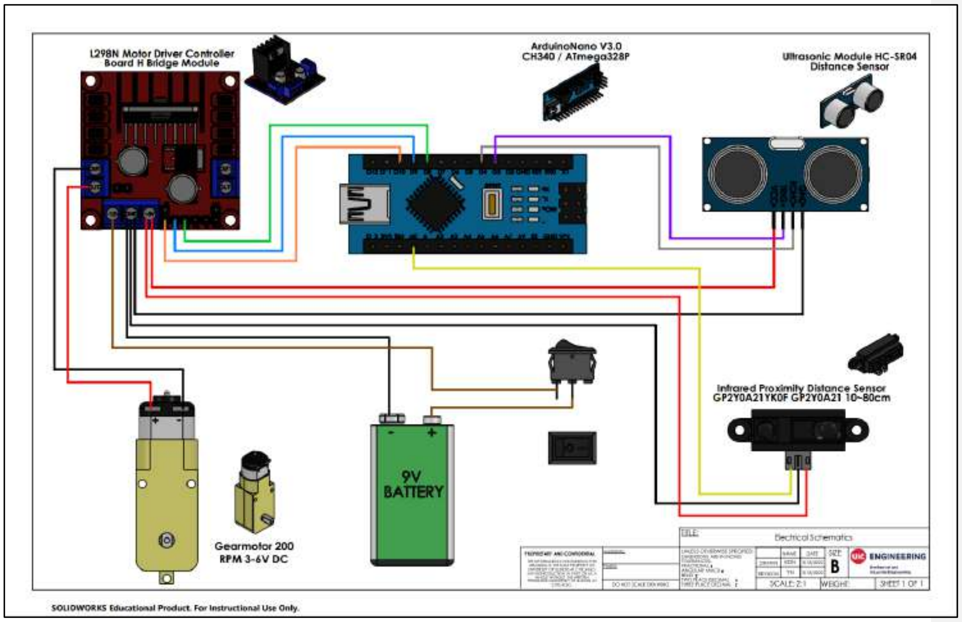


```python
    #Robot 2 is behind everything
    elif xPrev_2 < xPrev_1 and xPrev_2 < xPrev_3:
        velocity_1 = 0.4
        velocity_2 = 0.5
        velocity_3 = 0.4

        LagBot = LENGTH_ROBOT_1 + LENGTH_ROBOT_3
        LagVelocity = (velocity_1 + velocity_3)/2

        velocityTop = velocity_2 + ((velocity_1 + velocity_3)/2)
        tto = 2*LagBot / velocityTop

        xNew_1 = velocity_1 + xPrev_1
        xNew_2 = ((velocityTop * tto) + (LagVelocity * tto))
        xNew_3 = velocity_3 + xPrev_3

        xPrev_1 = xNew_1
        xPrev_2 = xPrev_2 + xNew_2
        xPrev_3 = xNew_3

        ttc = (abs(xPrev_2 - xPrev_1) - LENGTH_ROBOT_2) / abs(velocity_2 - LagVelocity)

        print("--Statement 2--")

    #Robot 3 is going faster
    else:
        velocity_1 = 0.4
        velocity_2 = 0.4
        velocity_3 = 0.5

        LagBot = LENGTH_ROBOT_1 + LENGTH_ROBOT_2
        LagVelocity = (velocity_1 + velocity_2)/2

        velocityTop = velocity_3 + ((velocity_1 + velocity_2)/2)
        tto = 2*LagBot / velocityTop

        xNew_1 = velocity_1 + xPrev_1
        xNew_2 = velocity_2 + xPrev_2
        xNew_3 = ((velocityTop * tto) + (LagVelocity * tto))

        xPrev_1 = xNew_1
        xPrev_2 = xNew_2
        xPrev_3 = xPrev_3 + xNew_3

        ttc = (abs(xPrev_1 - xPrev_3) - LENGTH_ROBOT_3) / abs(velocity_3 - LagVelocity)

        print("--Statement 3--")

```



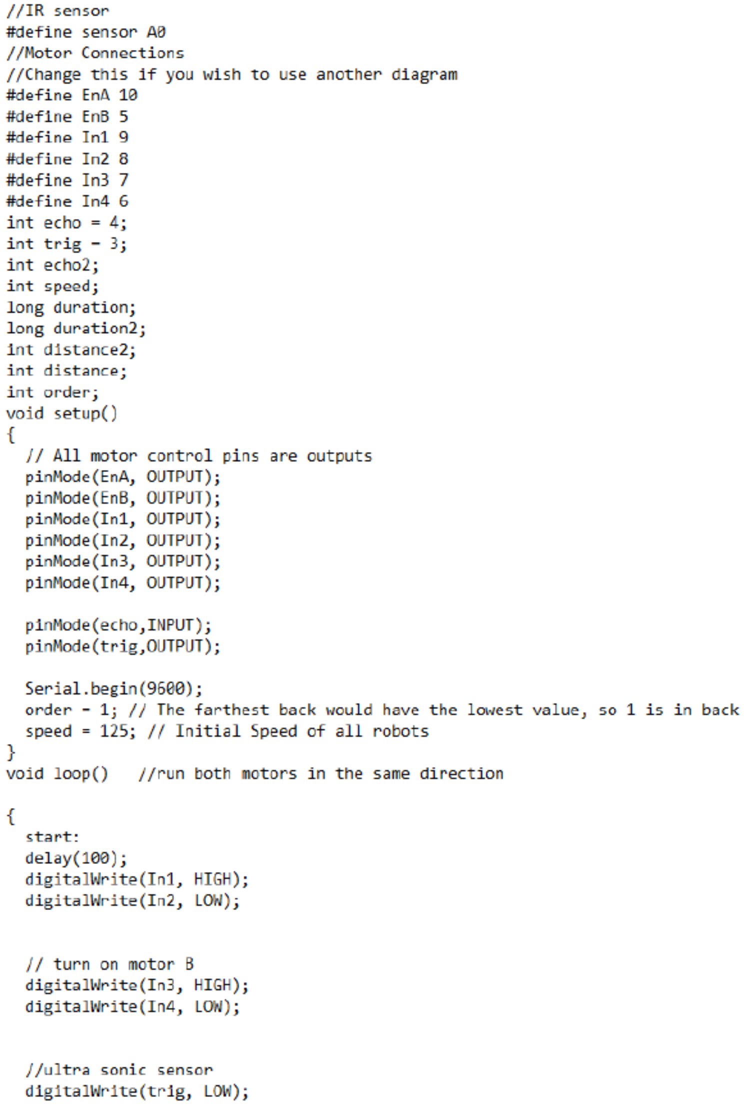
```python

    print("Current Position 1: " + str(xPrev_1))
    print("Current Position 2: " + str(xPrev_2))
    print("Current Position 3: " + str(xPrev_3))

    print("___________________________")


    t = t + tto + ttc

    tArray.append(t)

    #Fills the array for the graphs
    velArray_1.append(velocity_1)
    velArray_2.append(velocity_2)
    velArray_3.append(velocity_3)

    posArray_1.append(xPrev_1)
    posArray_2.append(xPrev_2)
    posArray_3.append(xPrev_3)

    #print('Time to taken to catch: ' + str(ttc))
    print('Time to taken: ' + str(t))

#Print the graphs
fig, (ax1, ax2) = plt.subplots(2)
fig.suptitle("Velocity/Position over Time")
ax1.set_title("Velocity (m/s)")
ax2.set_title("Position (m)")

ax1.plot(tArray, velArray_1)
ax1.plot(tArray, velArray_2)
ax1.plot(tArray, velArray_3)

ax2.plot(tArray,posArray_1)
ax2.plot(tArray,posArray_2)
ax2.plot(tArray,posArray_3)

plt.show()
```

Electronic Schematic with Arduino code



Arduino Code for 2 robot setup

Commented [5]: Update code



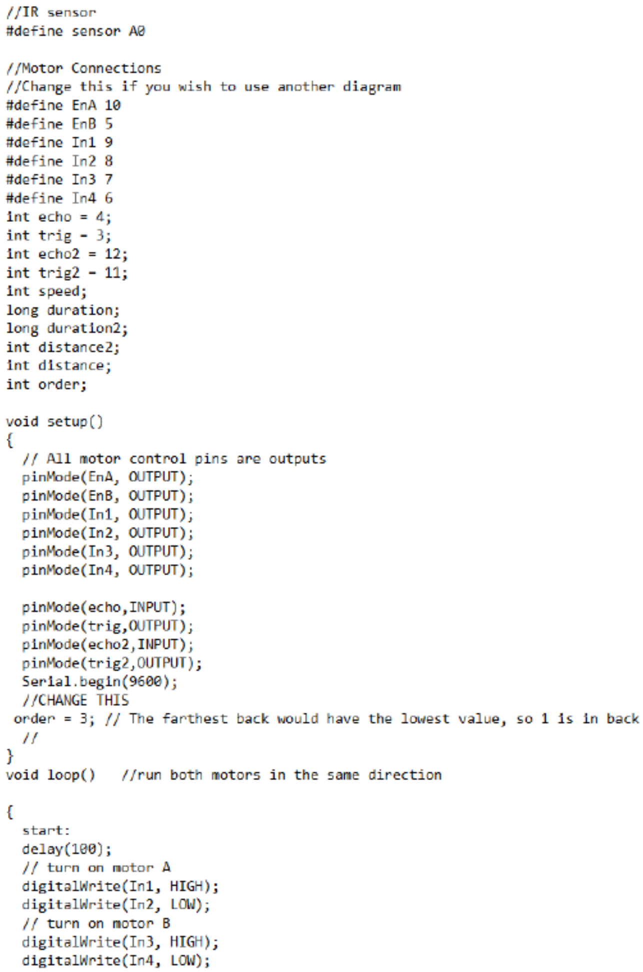
```
//IR sensor
#define sensor A0
//Motor Connections
//Change this if you wish to use another diagram
#define EnA 10
#define EnB 5
#define In1 9
#define In2 8
#define In3 7
#define In4 6
int echo = 4;
int trig = 3;
int echo2;
int speed;
long duration;
long duration2;
int distance2;
int distance;
int order;
void setup()
{
  // All motor control pins are outputs
  pinMode(EnA, OUTPUT);
  pinMode(EnB, OUTPUT);
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(In3, OUTPUT);
  pinMode(In4, OUTPUT);

  pinMode(echo,INPUT);
  pinMode(trig,OUTPUT);

  Serial.begin(9600);
  order = 1; // The farthest back would have the lowest value, so 1 is in back
  speed = 125; // Initial Speed of all robots
}
void loop()   //run both motors in the same direction

{
  start:
  delay(100);
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);


  // turn on motor B
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);


  //ultra sonic sensor
  digitalWrite(trig, LOW);
```

```
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance is: ");
  Serial.println(distance); // in cm

// IR sensor
  float volts = analogRead(sensor)*0.0048828125;  // value from sensor * (5/1024)
  int distance2 = 13*pow(volts, -1); // worked out from datasheet graph

   Serial.print("Distance is about: ");
      Serial.println(distance2); // in cm
  //      Serial.print("Order is ");
  //    Serial.println(order);

    analogWrite(EnA,speed);
    analogWrite(EnB,speed);
    delay(150);

   while (order == 1 && distance2 > 5)
{
   speed = 255;
    analogWrite(EnA,speed);
    analogWrite(EnB,speed);
  Serial.print("Catching up...");
  goto start;
}

  while (order == 1 && distance > 10 && distance2 < 5) // Should never have something on top in the back
  {
      order = 2;
      int speed = 255;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);
      Serial.print("Distance: ");
      Serial.println(distance); // in cm
      Serial.print("Order will be ");
      Serial.println(order);
      delay(4000); //Change based on speed
      goto start;
    }
```

```
  while (order == 2 && distance <= 10) // When back catches up
 {
     order = 1;
     speed = 255;
     analogWrite(EnA,speed);
     analogWrite(EnB,speed);
     delay(1500);
     speed  = 125;
     analogWrite(EnA,speed);
     analogWrite(EnB,speed);
     Serial.print("Distance: ");
     Serial.println(distance); // in cm
     Serial.print("Order is ");
     Serial.println(order);
     delay(500);
    goto start;
  }

while (int order = 2 && distance > 10
) // When it is not on top
  {
     order = 2;
      speed = 125;
     analogWrite(EnA,speed);
     analogWrite(EnB,speed);
     Serial.print("Distance: ");
     Serial.println(distance); // in cm
     Serial.print("Order is still ");
     Serial.println(order);

  goto start;
  }
  }
```

Arduino Code for 3 robot setup

Commented [6]: Update code



```
//IR sensor
#define sensor A0

//Motor Connections
//Change this if you wish to use another diagram
#define EnA 10
#define EnB 5
#define In1 9
#define In2 8
#define In3 7
#define In4 6
int echo = 4;
int trig - 3;
int echo2 = 12;
int trig2 - 11;
int speed;
long duration;
long duration2;
int distance2;
int distance;
int order;

void setup()
{
  // All motor control pins are outputs
  pinMode(EnA, OUTPUT);
  pinMode(EnB, OUTPUT);
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(In3, OUTPUT);
  pinMode(In4, OUTPUT);

  pinMode(echo,INPUT);
  pinMode(trig,OUTPUT);
  pinMode(echo2,INPUT);
  pinMode(trig2,OUTPUT);
  Serial.begin(9600);
  //CHANGE THIS
 order = 3; // The farthest back would have the lowest value, so 1 is in back
  //
}
void loop()   //run both motors in the same direction

{
  start:
  delay(100);
  // turn on motor A
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);
  // turn on motor B
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);
```

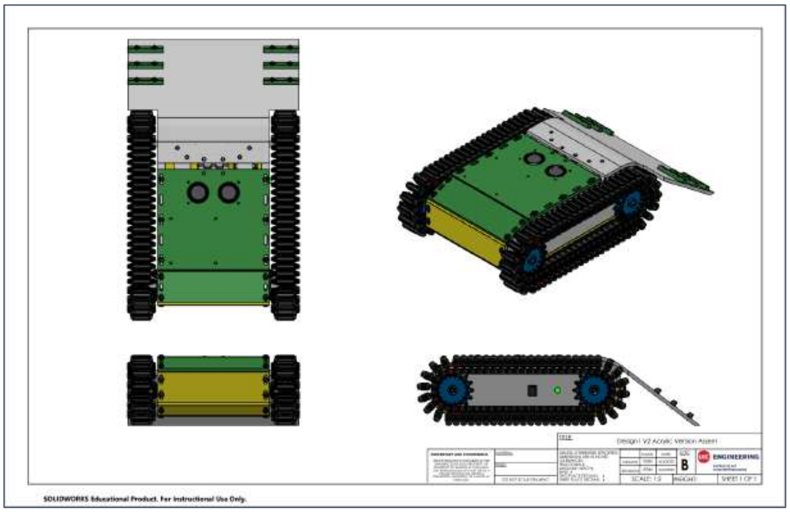
```
  // sensor
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = duration * 0.034 / 2;
  //Serial.print("Speed: ");
  //Serial.println(speed);
   Serial.print("Distance: ");
    Serial.println(distance2);

  float volts = analogRead(sensor)*0.0048828125;  // value from sensor * (5/1024)
  distance2 = 13*pow(volts, -1); // worked out from datasheet graph


while (order == 2 && distance <= 5) // When it is on top
  {
      order = 1;
      speed = 255;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);
      //Serial.print("Speed: ");
      //Serial.println(speed);
      delay(500);
      speed  = 175;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);

      Serial.print("Order is ");
      Serial.println(order);
      delay(2000);
       goto start;
   }
   while (order == 2 && distance2 > 5 )
{
  speed = 200;
  analogWrite(EnA,speed);
  analogWrite(EnB,speed);
  Serial.print("Distance: ");
  Serial.println(distance2);
  Serial.println("Catching up...");
  goto start;
}

   while (order == 1 && distance2 > 4)
{

    speed = 255;
    analogWrite(EnA,speed);
```

```
  // sensor
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = duration * 0.034 / 2;
  //Serial.print("Speed: ");
  //Serial.println(speed);
   Serial.print("Distance: ");
    Serial.println(distance2);

  float volts = analogRead(sensor)*0.0048828125;  // value from sensor * (5/1024)
  distance2 = 13*pow(volts, -1); // worked out from datasheet graph


while (order == 2 && distance <= 5) // When it is on top
  {
      order = 1;
      speed = 255;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);
      //Serial.print("Speed: ");
      //Serial.println(speed);
      delay(500);
      speed  = 175;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);

      Serial.print("Order is ");
      Serial.println(order);
      delay(2000);
       goto start;
   }
   while (order == 2 && distance2 > 5 )
{
  speed = 200;
  analogWrite(EnA,speed);
  analogWrite(EnB,speed);
  Serial.print("Distance: ");
  Serial.println(distance2);
  Serial.println("Catching up...");
  goto start;
}

   while (order == 1 && distance2 > 4)
{

    speed = 255;
    analogWrite(EnA,speed);
```

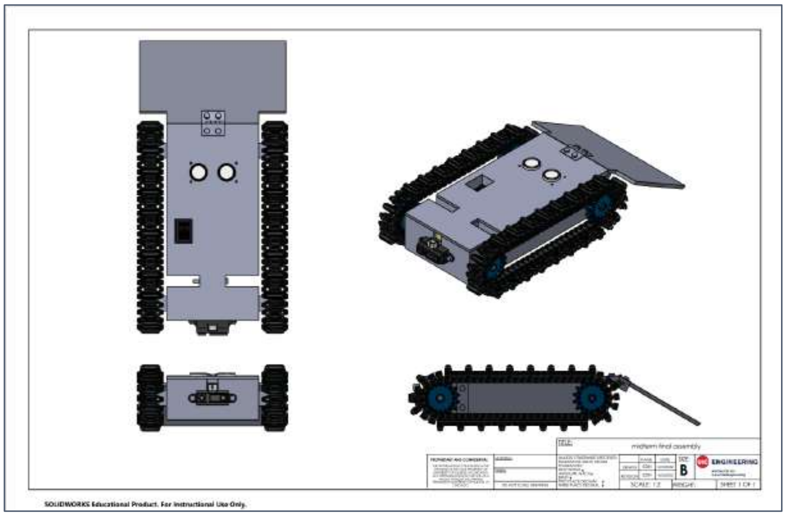
```
    analogWrite(EnB,speed);
    Serial.println("Catching up...");
     goto start;

}

  while (order == 1 && distance > 10 && distance2 < 5) // Should never have something on top in the back
  {
      order = 3;
      speed = 255;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);

      Serial.print("Order will be ");
      Serial.println(order);
      delay(2500); //Change based on speed
      goto start;
    }


    while  (order == 2 && distance2 <= 9)
  {
    speed = speed-15;
    analogWrite(EnA,speed);
    analogWrite(EnB,speed);
    Serial.print("Distance: ");
    Serial.println(distance2); // in cm
    Serial.println("Caught up...");
    goto start;
  }
   while (order == 3 && distance <= 10) // When it is on top
   {
      order = 2;
      speed = 255;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);
      delay(500);
      speed  = 175;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);
      Serial.print("Order is ");
      Serial.println(order);
      delay(2000);
     goto start;
   }
 while (int order = 3 && distance > 10 ) // When it is not on top
   {
      speed = 175;
      analogWrite(EnA,speed);
      analogWrite(EnB,speed);
       goto start;
```

Designs

Acrylic Design



Top View of PLC 3D Print Design



References

[0] - Garcia, Elena, et al. “The Evolution of Robotics Research.” *IEEE Robotics Automation Magazine*, vol. 14, no. 1, Mar. 2007, pp. 90–103. *IEEE Xplore*, https://doi.org/10.1109/MRA.2007.339608

[1] - “Robotics and Motion Laboratory @ University of Illinois at Chicago” Pranav A. Bhounsule, https://pab47.github.io/

[2] - “Robotics Laboratory College of Engineering.” University of Illinois at Chicago, https://robotics.ece.uic.edu/

[3] - News, In Compliance. “IEEE Published IEEE 1872-2015 Standard Ontologies for Robotics and Automation.” In Compliance Magazine, 10 Apr. 2015, https://incompliancemag.com/ieee-published-ieee-1872-2015-standard-ontologies-for-robotics-and-automation/

[4] - “Caterpillar Rolling Swarm Physical Demonstration” *YouTube*, uploaded by Ryan Dickerhoff, 14 Mar. 2018, https://www.youtube.com/watch?v=JMi20j0rkVg

[5] - Møller, Christian. *The Theory of Relativity*. Oxford University Press, 1994. ISBN 019560539X

[6] - Rosenthal, G.A.; Berenbaum, M.R. (1991). *Herbivores: Their Interactions with Secondary Plant Metabolites* (2nd ed.). Oxford: Elsevier Science. p. 190. ISBN 978-0-323-13940-3.

[7] - “Sawfly larvae increase their movement speed by using each other as a conveyor belt” *YouTube*, uploaded by Vinod Abraham, 8 Feb. 2021, https://www.youtube.com/watch?v=JMi20j0rkVg

[8] - Davis, Ben. “Home.” *MVOrganizing*, MVOrganizing, 14 Mar. 2021, https://www.mvorganizing.org/why-are-caterpillar-tracks-better-than-wheels/.

[9] - Gibbens, Sarah. “Watch This Creature's Bizarre Synchronized Wiggling.” *Animals*, National Geographic, 3 May 2021, https://www.nationalgeographic.com/animals/article/sawfly-larva-defense-amazon-video-spd.

[10] - Author Joe Boggs. “Is It a Sawfly Larva or a Caterpillar?” *BYGL*, 18 July 2019, https://bygl.osu.edu/node/1355.

[11] - News IC. IEEE Published IEEE 1872-2015 Standard Ontologies for Robotics and Automation. 2015. https://incompliancemag.com/ieee-published-ieee-1872-2015-standard-ontologies-for-robotics-and-automation/. Accessed Nov 12, 2021.

[12] - IEEE Standard Ontologies for Robotics and Automation, 1872 -2015.

[13] - Schranz, Melanie, et al. “Swarm Robotic Behaviors and Current Applications.” Frontiers, Frontiers, 1 Jan. 1AD, https://www.frontiersin.org/articles/10.3389/frobt.2020.00036/full.

[14] - DC gearbox motor - TT motor - 200RPM - 3 to 6VDC - digi-key. (n.d.). Retrieved March 1, 2022, from https://media.digikey.com/pdf/Data%20Sheets/Adafruit%20PDFs/3777_Web.pdf

[15] - L298n dual H bridge motor driver. Robotpark Global. (n.d.). Retrieved March 1, 2022, from https://www.robotpark.com/L298N-Dual-H-Bridge-Motor-Driver